KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

The Self-taught Speech Interface

Bart Ons

Supervisor:
Prof. dr. ir. Hugo Van hamme
Dr. msc. Jort Florent Gemmeke, co-supervisor

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering Science

20^{th} of May, 2015

The Self-taught Speech Interface

Bart ONS

Examination committee:
Prof. em. dr. ir. Yves Willems, chair
Prof. dr. ir. Hugo Van hamme, supervisor
Dr. msc. Jort Florent Gemmeke, co-supervisor
Prof. dr. ir. Marie-Francine Moens
Prof. dr. ir. Tinne Tuytelaars
Prof. dr. ir. Sharon Goldwater
(University of Edinburgh, United Kingdom)
Prof. em. dr. ir. Lou Boves
(Radboud University Nijmegen, The Netherlands)

Dissertation presented in partial
fulfillment of the requirements for
the degree of Doctor
in Engineering Science

20^{th} of May, 2015

Uitgegeven in eigen beheer, Bart Ons, Kasteelpark Arenberg 10 - box 2441, B-3001 Heverlee (Belgium)

ISBN

D/

Preface

Asking a machine to execute a job requires a second machine which translates human speech to appropriate clean symbols. This translation machine is called a vocal user interface (VUI). Since a VUI is a machine too, it prefers a clean consistent signal at its input. However, speech is an intricate pattern and humans are far from consistent in their pronunciations and word usage. Therefore, humans are required to adapt to machines by speaking in prescribed ways with clean articulated, consistent sounds and neatly ordered words.

We aim at designing a VUI that is able to listen to natural expressions. To this end, we developed models that are able to understand semantically relevant words in continuous speech. These word-semantic relations are acquired from the user as the user trains his own VUI. The challenge in this design is to endow the interface with machine learning mechanisms that enable learning and recognition of recurrently used expressions. This conduct of design should allow a user to choose his own words and expressions.

Endowing human-like understanding into a machine, although in a very primitive form, is a very attractive challenge and interesting subject to study. I'm grateful to have this opportunity. First and foremost, I would like to thank Hugo Van hamme and Jort Florent Gemmeke for all the prior work they put into the resourceful and original project proposal and the available Framengine software. This prior work gave my work a boost-start. I thank Jort for his day-to-day support and Hugo for his time to discuss ongoing research despite a tight agenda. I'm grateful to the other members in the ALADIN project for their constructive participation and to my colleagues for the numerous coffees and their companionship. I like to extend my gratitude to the members of my jury for their efforts and willingness to be in my jury and for their suggestions to improve the text in this dissertation. I also thank IWT (Agentschap voor Innovatie door Wetenschap en Technologie) to make this work possible (IWT-SBO grant 100049). Tot slot bedank ik mijn partner voor de steun en mijn jonge kinderen voor de motivatie die ik haal uit het zien opgroeien van hen.

Abstract

With advances in technology, human-machine interfaces have become commonplace. Their design requires a great deal of engineering efforts to make them functional and accessible. One of these engineering efforts is the embedding of voice control. This improves the accessibility for people with a physical disability. In common speech-enabled command-and-control applications, the spoken commands are restricted to a predefined list of phrases and grammars. These conventions work well as long as the system does not have to stray too far from the conditions considered by the designer or from the characteristics of the training material. Speech technology would benefit from training during usage; learning the specific vocalizations and the emerging expressions of the user. Designing a vocal user interface (VUI) model from this developmental perspective would widen accessibility and cater for users with non-standard or dysarthric speech.

The research in this dissertation is aimed at the development of a self-taught VUI that learns speech commands from the user while it is operational. To this end, we adopt and introduce different procedures in order to build a VUI-model that learns from a few learning examples. A learning example consists of two sources of information: the spoken command and the demonstration of the commanded action. Both sources of information are converted to fixed-length utterance-based vectors. The followed approach links the acoustic patterns that are embedded in the spoken utterances to the concepts that jointly define the meaning of the utterance. The method represents the data by its recurrent acoustic and semantic patterns and the incidence of these patterns in the data. Since these patterns are embedded in the data, the representation of the data has a significant influence over the performance of the VUI model. A thorough analysis of different representations resorting to speaker-dependent and speaker-independent data resources, is made. Attention is also given to the representation of the commanded action. The representation of the commanded action consists of an incidence vector representing the semantic content of the demanded action. Users are non-experts in training a VUI, therefore, errors

such as uttering an incomplete command or pushing a wrong button, will emerge. We demonstrate robustness against these kinds of errors. Another issue pertaining to semantics is the correlation between relevant concepts in a spoken utterance. This dependency is an additional source of information. We exploit this information and compare different semantic structures pertaining to these semantic dependencies.

With the focus on the learning process rather than on the resulting model, we develop procedures for incremental and adaptive learning. By exploiting a semi-Bayesian procedure called maximum a posteriori (MAP) estimation, the VUI model can be made to learn incrementally, one utterance at a time. Incremental learning procedures are developed at the level of the basic acoustic atoms and at the level of the word models. They are compared with their batch learning variants and yield comparable accuracy. The implementation of a forgetting factor makes the models adaptive to changes in the speech of the user.

The learning curves are an assessment of the quality of learning in function of the amount of training data. We analyse the learning curves for all these developments by numerous experiments in realistic learning scenarios implemented on computer. By this, we acquire a sense of the system's performance in a real-world training environment. The grounding of the VUI in its operational context and the training of the VUI by the user are the two most important key aspects that inspired the conception, the developments and the research in this dissertation.

Beknopte samenvatting

Door de technologische vooruitgang zijn mens-machine interfaces omnipresent in het dagelijks leven. Het bevorderen van de toegankelijkheid van een interface vraagt veel denkwerk en technische ontwikkelingen. Een voorbeeld van een technische ontwikkeling is het inbouwen van spraakherkenning. Dit bevordert de toegankelijkheid voor mensen met motorische beperkingen. In de meeste toepassingen met spraaksturing wordt het vocabularium en de zinsbouw van de spraakcommando's strikt gereglementeerd. Als mensen zich houden aan deze regels blijft alles behoorlijk werken, maar dit is vaak niet mogelijk voor mensen met spraakproblemen. Spraakherkenningstechnologie zou dan beter werken wanneer de uitspraak en de uitdrukkingen van de gebruiker aangeleerd worden. Het aanleren van de gebruikersexpressies hoeft niet hoofdzakelijk vooraf, maar kan ook tijdens het gebruik gebeuren. De gebruiker traint dan zijn eigen spraakinterface. Het ontwikkelen van een spraak interface vanuit dit ontwikkelingsperspectief zou de toegankelijkheid kunnen bevorderen voor mensen die dialect spreken of leiden aan dysartrie.

Het onderzoek spitst zich toe op het ontwikkelen van een taalverwervende spraak interface die leert van de gesproken woorden in de operationele context. Hiervoor werden procedures ontwikkeld die leren van voorbeelden. Een leervoorbeeld is het samen presenteren van het gekozen spraakcommando en de actie die daaruit moet volgen. De demonstratie en de spraak van een zin worden geformaliseerd in vectoren van gelijke dimensie. De gevolgde methode legt een verband tussen akoestische patronen van de gesproken zinnen en de concepten in die zinnen die tezamen de betekenis van de zin voorstellen. Omdat deze methode wederkerende patronen haalt uit de data is de formele voorstelling van deze data cruciaal. Deze voorstelling is gebaseerd op het voorkomen van kleinere akoestische patronen die geleerd worden aan de hand van data. De data kan afkomstig zijn van de gebruiker zelf of van een bestaande dataset met meerdere sprekers. Beide opties worden vergeleken en beschreven in experimenten die evolueren naar sprekersafhankelijke modellen. Aan de semantische representatie van de uitgevoerde acties wordt eveneens aandacht besteed. Deze representatie

is een vector die de aanwezigheid codeert van de semantische concepten die refereren naar de uit-te-voeren actie. De robuustheid tegen fouten in die representatie wordt onderzocht. Aangezien het voorkomen van deze semantische concepten correleren, kan deze correlatie gebruikt worden als een extra bron van informatie om betere herkenning te bereiken. We onderzoeken hoe de semantische structuur invloed heeft op het resultaat.

Aangezien we ons vooral richten op het leerproces in plaats van het model dat resulteert uit dit leerproces, ontwikkelen we procedures om incrementeel en adaptief te leren. Deze procedures faciliteren het leren van leervoorbeelden die incrementeel worden aangeboden. Een Bayesiaanse methode die de maximale waarde schat van de a posteriori verdelingen laat toe om incrementeel te leren. Deze procedure wordt toegepast op twee niveaus: het leren van de akoestische patronen en het leren van de woord modellen. De incrementeel lerende modellen benaderen de accuraatheid van de voorgaande modellen. Met de toevoeging van een vergeetfactor zijn ze ook adaptief voor veranderingen in het spraaksignaal of voor veranderingen van woordkeuze.

Leercurves worden opgesteld en tonen de kwaliteit van het leren in functie van de hoeveelheid data. Hiervoor worden experimenten uitgevoerd op computer die de spraakherkenning op een realistische wijze testen om zo inzicht te krijgen in de leerperformantie bij een reële implementatie. Het leren van betekenis uit de operationele context en het trainen van de modellen door de gebruiker zelf zijn de belangrijkste kenmerken die als leidraad dienen in de conceptie, het uitwerken en het uitvoeren van het onderzoek in dit doctoraat.

Abbreviations

ASR	Automatic Speech Recognition
ALS	Amyotrophic lateral sclerosis
CGN	Corpus Gesproken Nederlands
CVA	Cerebrovascular accident
DNN	Deep Neural Networks
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
ICA	Independent Component Analysis
HAC	Histogram of Acoustic Co-occurrence
LVCSR	large Vocabulary Continuous Speech Recognition
MAP	Maximum A Posteriori
MFCC	Mel Frequency Cepstral Coefficients
MIDA	Mutual Information Discriminant Analysis
ML	Maximum Likelihood
MLLR	Maximum Likelihood Linear Regression
MS	Modulation spectrogram
NMF	Non-negative Matrix Factorization
PCA	Principal Component Analysis
PLSA	Probabilistic Latent Semantic Analysis
SCI	Spinal Cord Injury
SNMF	Supervised NMF
SVQ	Soft Vector Quantization
UBM	Universal Background Model
VQ	Vector Quantization
VUI	Vocal User Interface

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With advances in technology and users relying on various applications, there is a growing need for accessibility to technological applications through user interfaces. Humans interact with technological devices by manual controls such as buttons, mouses, keyboards or touchscreens. These controls have become commonplace, but their designs are founded on the assumption that users have normal motor or visual ability. However, for people with a physical or visual impairment, manual controls are not always easy to handle. User interfaces are designed to be functional and pleasant to use, but the ongoing miniaturisation of mechanical and electronic devices are sometimes ergonomically inconvenient. Manual controls or displays are harder to fit on a small device and small keys are harder to hit. Voice control could be a viable solution to these design issues as its implementation is not hindered by the ongoing miniaturisation and as it provides handsfree control, thus catering for physically challenged people.

Traditional voice control such as voice dialling in cars or voice enabled home automation requires users to speak a phrase containing words from a predefined set and adhering to a predefined grammar. Hence, users have to adapt to the constraints of the vocal interface. Currently, natural language interfaces are integrated in voice enabled applications such as the voice controlled digital assistant in smartphones. These vocal user interfaces allow a great deal of flexibility in the way that users phrase their requests. However, these approaches are still language dependent and its availability is usually limited to widely spoken languages. Contrary to these approaches, we aim to develop voice control that widens accessibility to users with non-standard speech. Our approach is language independent since we aim to develop a vocal user interface (VUI) that learns the meaning, the grammars, the words and the vocalizations of the user.

The reported research in this dissertation was aimed at the development of a VUI that learns words and phrases from limited prior knowledge during its usage. In this introductory chapter, the problem statement is framed in a general scope that touches social, computational, philosophic and application issues. These issues were a guidance in the conducted research and the experimental designs in the subsequent chapters.

1.1 Social aspects

Offering comfort to people at home and at work, from coffee brewing on a coffee machine to lighting the house; all this is done by a simple button press. Since humans in Western societies grew up with these elementary luxuries, they are hardly aware how life would be without it. These features are firmly rooted in daily life, however, for people with a physical disability or a visually impairment, pushing a button on a keypad is not always an easy accomplishment. This is especially the case if people have loss of fine motor skills. Likewise, for people with reduced gross motor abilities, moving towards a light switch or a control panel requires quite an effort. For these groups, voice control is a viable solution. There is a growing consensus in health care that active and independent living is a priority for elderly people and people with a disability. Voice control is an aid that could support people in their regular daily routines, providing a significant improvement in the quality of life, their security and their communicative abilities. These aids could also yield profits for society as they open windows for people towards employment, help people to actively participate in society, and reduce health care service needs in housing and at work.

A large number of people in Flanders have reduced motor skills and could benefit from voice control. Causes of reduced motor control are numerous and have a significant prevalence in society. Based on prevalence numbers of diseases[1], many people would benefit from voice control. More than 50,000 people in Flanders have reduced motor or sensory control due to a cerebrovascular accident (CVA or stroke). The affected area of the brain does not function properly resulting in the disability to move one or more limbs. Occasionally, people have difficulties in understanding, speaking or seeing. A disease with a prevalence over 4,000 cases in Flanders is spinal Cord injury (SCI). It is an interruption of the afferent and efferent nerve structures leading to muscle weakness affecting lower limbs (paraplegia) or the whole body (tetraplegia). Motor, sensory and vegetative failures can occur. A third muscular disease is Amyotrophic Lateral

[1]The numbers were taken from the ALADIN project proposal [1] in which the prevalence was rudimentary investigated.

Sclerosis (ALS), which leads to rapidly progressive muscle weakness. Symptoms are dysphagia and respiratory problems to name a few. Its prevalence is low which is about 400 in Flanders. Multiple sclerosis (MS) is an auto-immune disorder affecting the central nervous system and it takes several forms, with new symptoms developing (progressive forms) or occurring in short periods (relapsing forms). The symptoms may include periodically reduced speech or degenerative speech ability. The prevalence of MS is about 9,000 patients in Flanders. With a prevalence of about 15,000 persons in Flanders, Parkinson's disease is the third most prevalent neurological disease following CVA and Alzheimer's disease. The impact of Parkinson's disease is wide ranging and affects mobility and occasionally speech intelligibility. Parkinson's disease is associated with ageing. By contrast, cerebral palsy (CP) is caused by damage to the motor control centres of the developing brain and occurs during pregnancy or at birth. It causes physical disability often co-occurring with spasticity and problems with sensation and communication abilities. The prevalence is about 10.000 in Flanders. An even larger potential user group are elderly people facing difficulties with motor control. Obviously, a large group of people could benefit from voice control.

These rudimentary prevalence numbers indicate that vocal and motor impairments often co-occur. Motor speech disorders resulting from neurological diseases or neurological injury is called *Dysarthria*. It is characterized by poor articulation due to articulatory muscle weakness. The production of speech is impaired while cognition and language understanding are intact. Besides articulation, other speech production subsystems such as respiration and prosody can be affected too. Thousands of people would benefit from voice control if voice control technology could master several levels of intelligibility. If speech recognition technology adapts to user's reduced speech ability, accessibility would widen and voice-enabled application could cater for people with non-standard speech.

There is a wide range of applications to which voice control adds value. Examples are command-and-control in assistive technology (see [2–5]) and automation in home and residential environments, such as opening curtains, windows, doors and shutters or lighting the house. Voice control is also used to remotely control T.V., radio or to move through menu's on a PC or a smart phone. It can facilitate access to social media and computer entertainment. Voice control could be used to adjust a hospital bed, to steer a wheelchair or to control a hoist. When people are immobilized after a fall, it facilitates an emergency call. Other applications are automatic call processing in telephone networks and query-based information requests providing updated travel information, news or weather reports. Although there is a clear market and a considerable economical potential, speech recognition technology only partially fulfils these

opportunities with partial success. The reason for this partial fulfilment is that speech interfaces are not always user-friendly and that automatic speech recognition (ASR) is not always working properly. This is especially the case when users speak in a natural person-to-person style or when they speak in non-standard ways such as dialects or accents. In the next section we describe state of the art automatic speech recognition (ASR) and typical approaches that alleviate ASR problems with non-standard speech.

1.2 State of the art ASR and adaptation

An important historical contribution to speech recognition was made at AT&T Bell Laboratories in the early eighties where services to the public, such as voice dialling and command-and-control applications for phone call routing were developed. The objectives of the research program was to create one system that caters for many different users without the need for individual speaker training. Thus the focus at Bell Laboratories was a speaker-independent system that was capable to manage acoustic variability coming from different speakers and that was equipped with prepared acoustic models to provide immediate usability (see [6]). This focus led to the creation of word and speech sound clustering algorithms that initially operated on templates but eventually developed into statistical models. These models coped with all sorts of variability caused by differences in accent, speaking speed or vocal tract length. It ultimately led to the introduction of mixture density hidden Markov models [7–10] (HMM). HMMs involve two nested distributions, one pertaining to the Markov chain which is a probabilistic transition of states, and the other one pertaining to a set of emission densities modelling the probability of physical observations. Each observation is associated with a state of the Markov chain. The standard emission densities in the HMM are modelled using Gaussian mixture models (GMM). A GMM is a convex linear combination of multivariate Gaussian probability distributions. The HMMs were able to capture an exhaustive part of speech variability and accents and they were further improved by the use of cepstrum, first and second order derivatives [11], the use of context-dependent phones [12], n-gram language models [13], and the Baum-Welch optimisation algorithm [14]. Currently, Deep Neural Networks (DNN) with many layers are a viable alternative to GMM emission probabilities (see [15]).

However, these initial objectives at AT&T Bell Laboratories left their mark on state-of-the-art ASR today. ASR was developed for immediate usability using speaker-independent models. As a consequence ASR performance degrades quickly for language variations or non-standard vocalizations that do not match the training material. Therefore, adaptation procedures were developed that are

capable adapting the acoustic model to the deviating speech. Two adaptation methods that requires limited adaptation data are *maximum a posteriori (MAP) adaptation* [16] and *maximum likelihood linear regression (MLLR)* [17].

In MAP adaptation, the mode of the posterior density of the GMM parameters is sought as follows,

$$\theta_{MAP} = \arg\max_{\theta} f(\mathbf{X}|\theta)g(\theta). \tag{1.1}$$

The function f is the likelihood of the speaker-dependent data $\mathbf{X}$ given the GMM parameters θ. The function g is the prior density of the GMM parameters. It is usually a product of a Dirichlet distribution and a Gamma-Normal distribution with hyperparameters chosen in accordance with the speaker-independent GMM parameters and the desired rate of adaptation. We will extensively deal with MAP-estimates in **Chapter 6** for the purpose of online learning.

In MLLR, a linear transformation is estimated in order to adapt Gaussian means and covariances to a specific speaker. If we denote the pre-trained Gaussian means and covariances by μ_{SI} and Σ_{SI} with the subindex 'SI' denoting the abbreviation 'speaker-independent', then adapted Gaussian means and covariances are obtained by the following transformations

$$\mu_{SA} = \mathbf{W}\mu_{SI} + \mathbf{b}, \tag{1.2}$$

$$\Sigma_{SA} = \mathbf{A}\mathbf{\Sigma}_{SI}\mathbf{A}. \tag{1.3}$$

The subindex 'SA' denotes 'speaker-adapted'. The matrices $\mathbf{W}$ and $\mathbf{A}$ are square linear transformations with their dimensions equal to the covariance matrix. The column vector $\mathbf{b}$ is a translation in the feature space. In MLLR, a linear relationship is assumed between the means in Equation 1.2 and a quadratic relationship between the covariances in Eq. 1.3. Since the number of parameters in this model is less than the number of GMM parameters, the linear transformation can be estimated from a limited amount of speaker-dependent data.

If adaptation data resources are limited, only means are adapted or the same transformation is presumed for the covariance matrix, thus $\mathbf{W} \equiv \mathbf{A}$. If a sufficient amount of adaptation data is available, MLLR can be extended to multiple piecewise transforms. Other techniques such as eigenvoices [18] could be feasible alternatives. However, this would require enrolment data from many speakers which is not trivial for disordered speech. Moreover, disordered speech contains more pronunciation variability ([19]) than normal speech, so the risk that the eigenvector method is not able to model adequately a new dysarthric speaker is a concern. Therefore, MLLR or MAP procedures using limited speaker-dependent data are common procedures in projects handling dysarthric speech. For example, the *VIVOCA2* (Voice Input Voice Output Communication

Aid, http://www.sheffield.ac.uk/cast/projects/vivoca2) project aims to produce a device that recognises dysarthric speech and produces clear synthesised speech which can be understood by any listener. *SPECS* (Speech-driven Environmental Control Systems, http://www.sheffield.ac.uk/cast/projects/specs) is a project of the same research unit in the University of Sheffield. The aim is to develop an environmental control system (ECS) [2, 5] using adaptation and noise robustness techniques in automatic speech recognizers. They target disabled and elderly people, often paired with speech disorders such as dysarthria. A third example is *HomeService*, which involves voice-enabled assistive technology in the home using cloud-based ASR [20].

Besides the availability of speaker-dependent adaptation material, projects dealing with dysarthric speech have many complications. Development and implementation costs are high as speech pathologies are ranging widely and targeted user groups are rather small. Additionally, user requirements are diverse and some user groups might need personal assistance or a tailored system. ASR requires users to adapt to the apparatus instead of the other way around and adaptation procedures often fail to adapt ASR to severe dysarthric speech. Moreover, from a user's perspective, their voice may change over time due to the progressive nature of some diseases and these changes require blind unsupervised adaptation. In different words, an exhaustive effort in the field of ASR research is required to develop good techniques that grant reliable usability and accessibility to various groups. The research in this dissertations forms part of the ALADIN project. In the ALADIN project we aim to provide an answer to the shortcomings of ASR and adaptation with a strong focus on non-standard speech such as dysarthric speech.

1.3 The research context

Contrary to the adaptation approach, the basic approach in the ALADIN project is to build a VUI model that learns from speech and demonstrations of the end user without any form of transcription. In this language-independent approach, spoken commands are learned by mining the speech input from the end user and the changes that are provoked on a device during a demonstration of the meaning of a command. The VUI infers its own annotations from signals referring to content and context information by reading the internal states from the devices. The aim of the ALADIN project is to develop, to evaluate, to demonstrate a new technology and to make it attractive to third parties that are prepared to implement the technology or to invest in the social benefits of the project.

Realising the ALADIN project requires a cooperative effort between many institutions that pooled their strengths and specialisms in various domains. One of those domains is *usability*. Usability deals with the characteristics of the user group such as their abilities, their limits and their preferences. This research also includes userfriendly protocols to switch the ALADIN system in a learning mode when demonstrations are given to the system or to indicate a correction when a wrong command was executed. To this end, a virtual home automation environment was set up and a graphical interface was developed on a tablet taking into account the physical challenges of the target group. The major part of this work was done by CUO (Centre for User Experience Research) at KU Leuven ([21–24]).

Another item on the to-do list deals with grammatical issues. The sequential order of the words pertains to the meaning of the words. For example, "put A on B" is not the same as "put B on A". The ALADIN system has to learn the grammatical structure of the commands that the user prefers. *Grammar induction* was done by training a HMM for each command. The HMM has a left-to-right state sequence to represent the chronological order of the spoken words and its emission densities are related to word-based acoustic representations. These word-based acoustic representations are obtained from the word learning models that are developed in the subsequent chapters of this dissertation. The grammar induction models were tested by using the PATCOR database. It is a collection of spoken commands from users playing the card game patience on computer. This card game contains commands such as "Put the four of clubs on the five of hearts" for which grammar induction is a real challenge. The major work in this part was done by the department of Computational Linguistics & Psycholinguistics (CLiPS) at the university of Antwerp ([25–28], see also [29]).

A vocal user interface has to deal with environmental noise when family members of the user are talking in the background, when music is playing on the radio or the television or when doors are slapped. Moreover, a mobile voice-enabled device has to deal with changing room acoustics and should be able to generalise the learned commands in different environments. AdvISE (Advanced Integrated Sensing) is dealing with *Noise robustness*. They investigate the use of microphone arrays and speech enhancement [30].

Mobilab (Expertisecentrum welzijn en Technologie) has dealt with demonstrators such as the implementation of a vocal command-and-control home automation system in a virtual 3D environment and a vocal assistant to command and control the TV.

The ALADIN system learns from demonstrations and for this, a second interface is needed to demonstrate the action that the user intends to activate with his spoken command. This second interface is controlled by hand and requires

the physical ability to push touchscreen buttons and to go through menu's. A manual interface is not always suited for the targeted user group and this contradicts the need of a vocal user interface. Therefore different procedures have been investigated to alleviate the dependency on manual control. One of these procedures was the use of a closed microphone that opens while pushing a button. This requires a great deal of timing and time-dependent coordination. A better choice was to use an open microphone that wakes up the ALADIN system by means of a wake-up word. This procedure is combined with a scanning interface that sequentially highlights the menu alternatives on the display. The user indicates his choice by speaking a specific word during the highlighting of the menu option. There are no reported studies on this topic but all these mechanisms were finetuned in between various demonstrations.

CLIPS and MOBILAB collected speech recordings that targeted the two voice-controlled applications. One data collection contained recordings of commands enabling handsfree gaming on computer. The other collection contained typical commands in a home automation setting. These recordings were used in **chapter 5** and **chapter 6**.

The department ESAT at the KU Leuven focussed on the acoustic models and the vocabulary acquisition. More in particular, we focused on learning to understand the meaning of self-chosen commands and learning from dysarthric speech. For this purpose novel techniques had to be developed for discovering correspondences between highly variable stretches of speech and what is conventionally considered as a "word". The chapters in this dissertation address topics that concern these fields. Unfortunately, at the start of the ALADIN project no recordings of dysarthic speech were available. However, it was possible to investigate a number of basic mathematical and algorithmic problems that had to be solved using an existing corpus containing normal speech. For this reason the research described in **chapters 2, 3 and 4** were based on corpora containing normal speech. However, in chapters **chapter 5 and 6** it will become clear that the results obtained by analysing normal speech directly feed into one of the main tasks: learning to understand the meaning of dysarthric speech. In the following sections, we explain the research goals and we focus on vocabulary learning, the followed approach and the motivation of this choice.

1.4 Problem statement

1.4.1 Knowledge sources

State of the art ASR applications make use of three knowledge sources. The first one is the acoustic model. Most systems employ the HMM-based three-

state context-dependent phone model with GMM emission densities. A second knowledge source is the lexicon which is an exhaustive list of all words and their pronunciations with phone symbol sequences. The third knowledge source is a language model or grammar that constrains the order in which words are composed to form an expression. For limited vocabulary applications, finite state grammars are commonplace, whereas statistical n-gram models are more convenient for large-vocabulary continuous-speech recognition (LVCSR). The three knowledge sources are usually built prior to usage and remain static from beginning to end. In speaker-independent applications, they are specified by the application developer. The three knowledge sources are language dependent and the lexicon and the grammars are usually also application-dependent. The current attitude to speaker-adaptation refers to adaptivity that is usually limited to the acoustic model and requires the user to read a text in order to collect speaker-dependent training material. These standard procedures fall short for severe dysarthric speech.

In the ALADIN approach, we make use of an additional knowledge source, namely, the mapping from acoustics to semantics. This knowledge source is obtained from the user by mining his spoken commands and his demonstrated actions. This method does not require transcriptions or word models and allows a language-independent learning of the spoken commands. The VUI infers its own annotations from signals referring to content and context information by reading the internal states from the devices. Obtaining this information is application-dependent and concerns the implementation of the system. This matter is out of the scope of this dissertation. We circumvent this problem by assuming that semantic descriptions are obtained from *grounding*. The grounding process [31] refers to the process by which common ground or meaning is built between the user and the system while they are interacting with each other. Spoken commands make sense if they anticipate the behaviour of the VUI in the environmental and operational context of the VUI. We assume that VUI actions make sense if the VUI learns to understand classes referring to devices, their actions, properties and states in the operational context.

1.4.2 Dysarthric speech

The ALADIN approach learns the meaning of spoken commands during usage. This makes it suitable to learn from dysarthric speech. Other approaches employ adaptation procedures and fall short for severe dysarthric speech. A comparison between dysarthric speech and normal speech is depicted in Figure 1.1. Spectrograms of five repetitions of the same phrase are tiled vertically on the left and right side. The five spectrograms on the left side are dysarthric spoken utterances from the same speaker. The five spectrograms on

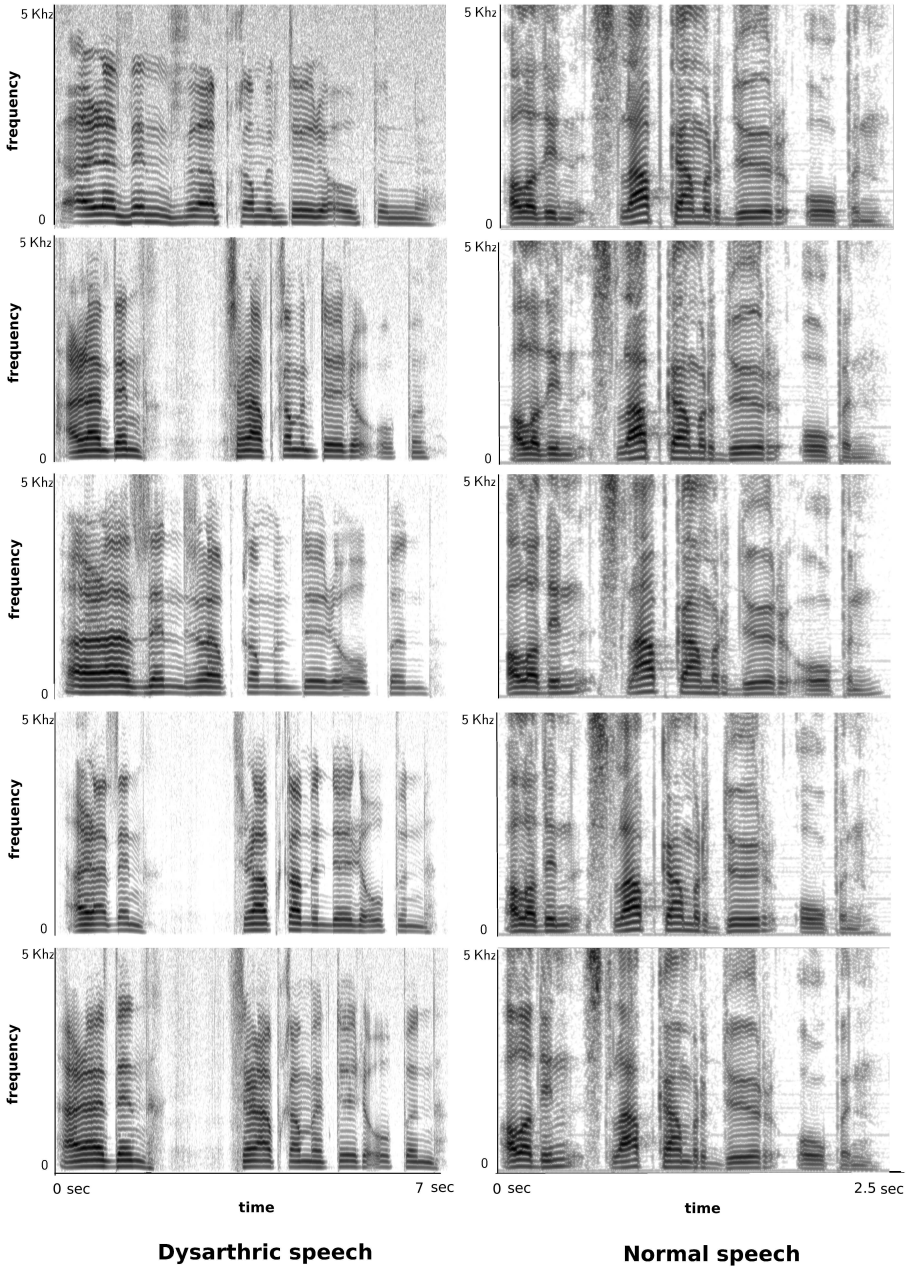


Figure 1.1: *Spectrograms of five repetitions of the phrase "Aladin, slaapkamerdeur open" for a speaker with severe dysarthria (intelligibility score of 66.1 following the procedure in [32]) at the left and normal speech at the right.*

the right side is normal speech from the same speaker. Both speakers uttered the phrase "Aladin, slaapkamerdeur open" which translates to English as "Aladin, open the bedroom door". The depicted frequency range is from 0 to 5 kHz. Darker texture corresponds with Higher spectral energy. Note that the spectral energy of the dysarthric speech examples is low at high frequencies whereas the normal speech spectrograms demonstrate substantial energy at 5 kHz. A second difference is the consistency of the signal. The spectrograms are very similar for normal speech, but less so for dysarthric speech. Although some similarities between formants at the left are traceable, there is a non-uniform rhythm with varying pauses in between. The total time is more or less 7 seconds for the dysarthric speech examples and 2.5 seconds for the normal speech examples on the right. Although the dysarthric speech examples share some patterns, the correspondence between the dysarthric and normal speech is hard to trace. This gap impedes adaptation from standard to dysarthric speech. This observation suggests that a speech recognizer that learns the mapping from acoustics to semantics is more feasible.

A second adaptation problem is related to the degenerative nature of some diseases. Speaker adaptation is usually carried out with a single adaptation episode before the use of the voice-enable application. One adaptation episode might be inadequate given the degenerative nature of some diseases. Adaptation during usage could improve usability in the long run. To some extent, the shortcomings of adaptation procedures are rooted in the traditional ASR aim to build a speaker-independent, lowcost and off-the-shelf ASR component. Adaptation procedures were built to alleviate the incompliance of typical ASR. Contrary to these adaptation approaches and off-the-shelf ASR, we aim at building a VUI that learns during usage, that learns the words from the user in the environment of the user. Our approach makes it possible to explore and capture the distinctive features during usage no matter whether user speech is normal, dialect or dysarthric. We start from a minimal amount of prior knowledge and try to develop a speaker-dependent system that continuously learns from the user. Such an approach would facilitate adaptation to degenerative speech abilities of particular user groups. However, the downside of this approach is that we should envisage application with a realistic lexicon size within the scale of a few hundred words. Dealing with larger lexicon sizes is outside the scope of this study.

1.4.3 Learning semantics

The semantic content is represented by frames, slots and slot values ([33]). Slot values composes the semantic indivisible units in the spoken utterance. These values refer to classes such as properties, actions and object names of the

controlled devices. The frame description holds the nested structures of these classes. We refer to **chapter 5** for a detailed description of frame structures and its applied examples.

Multiple slot values are active in a single utterance and confine the learning problem to a multi-label classification problem. The problem is further compounded by the lack of word boundaries and semantic time annotations. Thus a collection of active semantic entities is associated with a collection of utterance-based acoustic features. This implies that one-to-one relations should be inferred from accumulating statistical evidence across different utterances and modalities in order to find statistical co-occurring regularities in the multimodal input. In this dissertation, we assume that the input of the devices is given and clearly definable by a binary vector indicating the presence and absence of the slot values. Since slot values refer to properties, actions and object names, their associated acoustic regularities are plausibly related to spoken words. Therefore, we refer to this learning by the term "vocabulary acquisition". The obtained wordlike units can be used in the grammar induction procedures. Grammar induction provides an additional source of information that facilitates the inference of the utterance-based semantic content.

1.4.4 The research goals

A different approach to learning speech and speech understanding leads to specific goals. In this research, we pursuit the following goals:
(I) To design a vocal interface that learns the semantics from the user.
(II) To design a system that learns from a few learning examples. One learning example consists of a spoken command and a demonstration of the action on the commanded device. Since the users make tutoring efforts, learning should be fast in the sense that sufficient accuracy is obtained from a few examples.
(III) To design a system that handles non-standard speech. The user's speech can be dialectal or pathological to a varying degree. The vocal user interface should be able to learn from dysarthric speech.
(IV) To design a system that learns from its operational context, starting from a tabula rasa or limited knowledge and developing its acoustic, lexical and grammatical models during usage. This goal would guarantee full adaptivity in the sense that no prior knowledge remains or should be unlearned.

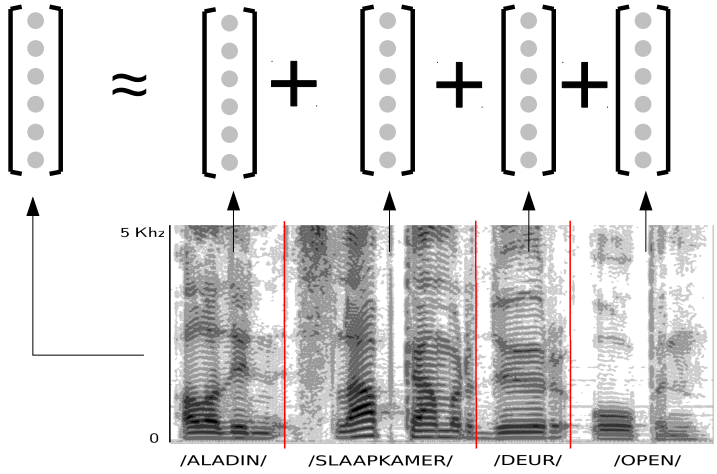


Figure 1.2: *Additivity of the acoustic and the semantic features. The feature representation of the utterance "Aladin, slaapkamerdeur open" is approximately equal to the sum of the feature representations of the separate words in the utterance.*

1.5 The approach

1.5.1 Data representation

The vocabulary acquisition follows a data mining approach that finds recurring acoustic patterns and relates them to semantic descriptions. Hereto, the acoustic input is mapped onto a fixed-dimensional representation, irrespective of the length of the utterance or the speech fragment. Hence both sentences and words get a representation in the same space. Word and sentence semantics are also mapped onto a fixed-dimensional representation. An essential property of both mappings is linearity: the representation of the whole (utterance) is the sum of the representation of the parts (words). The linear additivity is depicted in Figure 1.2. Linearity is exploited by finding the representations of the parts through non-negative matrix factorization (NMF) of the representations of many utterances and their semantics. We now elaborate on these concepts.

The complete list of slot values is represented by a fixed-length binary column vector, denoted by $\mathbf{v}_s^{(n)}$ with subindex s indicating that $\mathbf{v}_s^{(n)}$ composes the

semantic part of the utterance-based vector. n is an utterance-dependent index. The presence or absence of slot values is indicated by ones and zeros.

Likewise, a vector is composed with acoustic information. The acoustic information of utterance n is denoted by $\mathbf{v}_a^{(n)}$, with subindex a indicating that $\mathbf{v}_a^{(n)}$ composes the acoustic part of the utterance-based vector.

The feature vector $\mathbf{v}_a^{(n)}$ is built in two steps. In the first step, the spectrographic representation is transformed into a posteriorgram. Given the frames of the spoken utterance and a set of exclusive basic acoustic models, the posterior probabilities that the frame observations are drawn from these exclusive basic models is presented in a two dimensional data structure. This structure is called a posteriorgram. The basic models can be phone-HMMs or Gaussians inferred from clustering procedures such as k-means (see **chapter 3**). In the second step, a histogram of acoustic co-occurrence (HAC, [34]) is build. HAC representations are utterance-based vectors and contain the co-occurrence statistics of the basic acoustic events. This co-occurrence statistics are accumulated over the whole utterance with a fixed time lag in between the frames, giving them the desired linearity property (See Figure 1.3).

If t denotes the time dependent index of the frame denoted by $\mathbf{x}$ in the n^{th} utterance spanning Q frames, the co-occurrence probability over a time lag τ for two basic acoustic atoms θ_a and θ_b, is defined as follows,

$$[\mathbf{v}_n^{\tau}]_{(\theta_a,\theta_b)} = \sum_{t=0}^{q-\tau} \mathbf{P}\left(\theta_a|\mathbf{x}_t\right) \mathbf{P}\left(\theta_b|\mathbf{x}_{t+\tau}\right). \tag{1.4}$$

with $\mathbf{P}\left(\theta|\mathbf{x}_t\right)$ the posterior probability ($\sum_{\theta} \mathbf{P}\left(\theta|\mathbf{x}_t\right) = 1$) that an θ-event occurred at time t.

The HAC accumulates all the co-occurrence probabilities in one vector with respect to a particular time lag τ. Multiple time aspects are incorporated by stacking multiple HAC's with shorter and longer delays. As a result, a large fixed-length vector is built that we denote by $\mathbf{v}_a^{(n)}$.

Given a training set $\mathbf{T} = (\mathbf{v}_a^{(1)}, \mathbf{v}_s^{(1)}), (\mathbf{v}_a^{(2)}, \mathbf{v}_s^{(2)}), \ldots, (\mathbf{v}_a^{(N)}, \mathbf{v}_s^{(N)})$ a data matrix $\mathbf{V}$ is composed by concatenating the the respective column vectors $\mathbf{v}_s^{(n)}$ and $\mathbf{v}_a^{(n)}$,

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_s \\ \\ \mathbf{V}_a \end{bmatrix} \tag{1.5}$$

with $\mathbf{V}_s = \beta \left[\mathbf{v}_s^{(1)}\mathbf{v}_s^{(2)}, \ldots, \mathbf{v}_s^{(N)}\right]$ and $\mathbf{V}_a = \left[\mathbf{v}_a^{(1)}\mathbf{v}_a^{(2)}, \ldots, \mathbf{v}_a^{(N)}\right]$. The weight factor β is a positive scalar balancing the relative importance of the recurring

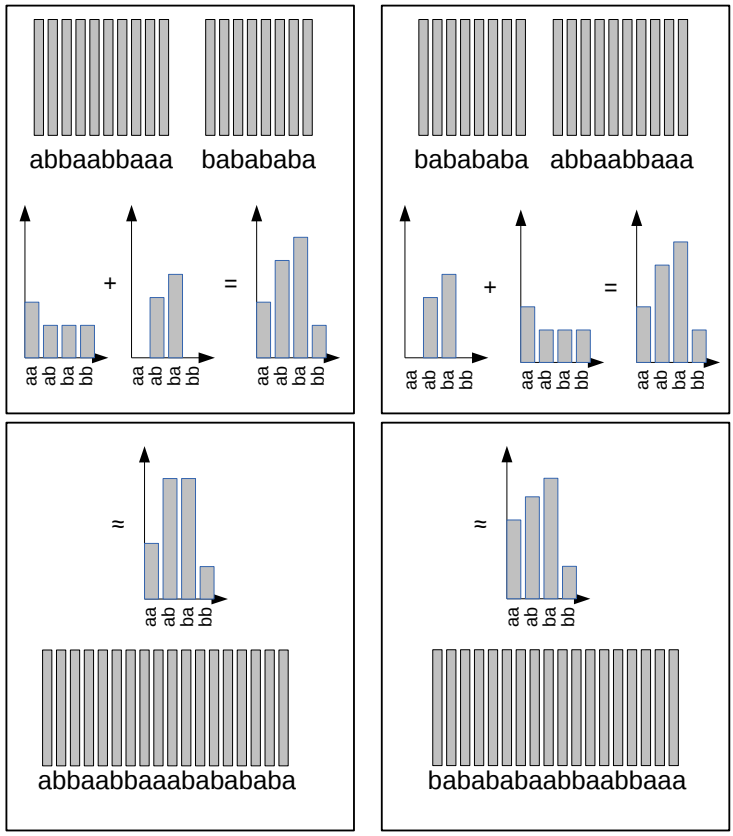


Figure 1.3: *The additivity property is approximately obeyed. The upper panels present the HAC features of the same two words. The chronological order is opposed in the left and the right panel. The resulting diagram in these panels contain the sum of the word-based HACs. In the lower panels are depicted the HAC diagrams of the utterances composed of the two words in the respective upper panels. The summed and the utterance-based HAC diagrams are all very similar to each other. Small count differences are caused by the missing counts at the word boundaries and the utterance boundaries. These missing counts become insignificantly small if words are spread over many frames.*

semantic and acoustic data patterns. The data entries in $\mathbf{V}$ are constrained to be real-valued non-negative numbers. This is the case if data entries consist of energetic magnitudes, probabilities or counts. A common practice is to choose

β so that the L1-norm of $\mathbf{V}_s$ and $\mathbf{V}_a$ are matched.

1.5.2 Non-negative matrix factorization

Earlier studies have demonstrated that joint non-negative matrix factorisation (NMF) is a useful tool to learn co-occurring regularities over a single or multiple modalities. An example is NMF-based keyword learning [35]. The data matrix $\mathbf{V}$ is factorized in the product of two lower rank matrices $\mathbf{W}^{\in\mathcal{R}^{F\times R},\geq 0}$ and $\mathbf{H}^{\in\mathcal{R}^{R\times N},\geq 0}$ as follows,

$$\mathbf{V} \approx \mathbf{WH} = \begin{bmatrix} \mathbf{W}_s \\ \\ \mathbf{W}_a \end{bmatrix} \mathbf{H}, \tag{1.6}$$

where the inner dimension R of $\mathbf{W}$ and $\mathbf{H}$ is usually chosen such that $R \ll N$.

The most appropriate loss function depends on the statistical structure of the data matrix. Assuming that entries in $\mathbf{V}$ are counts of events, an appropriate loss function for the approximation in Eq. 1.6 is the generalised Kullback-Leibler divergence (gkld) or I-divergence:

$$D_{KL}(\mathbf{V}||\mathbf{WH}) = \sum_{i=1}^{F}\sum_{n=1}^{N}\left[v_{in}\log\frac{v_{in}}{[\mathbf{WH}]_{in}} - v_{in} + [\mathbf{WH}]_{in}\right] \tag{1.7}$$

with $[.]_{in}$ denoting the i^{th} row and n^{th} column entry of the matrix inside the brackets. It was shown that this loss function is based on the Poisson noise assumption and fits well for count data [36].

Lee and Seung [37] presented alternating multiplicative update rules for minimizing Eq. 1.7 with respect to the entries h_{rn} in $\mathbf{H}$ and w_{ir} in $\mathbf{W}$:

$$h_{rn} \leftarrow h_{rn}\frac{\sum_{i=1}^{F}\frac{v_{in}}{[\mathbf{WH}]_{in}}w_{ir}}{\sum_{q=1}^{F}w_{qr}}, \tag{1.8}$$

$$w_{ir} \leftarrow w_{ir}\frac{\sum_{n=1}^{N}\frac{v_{in}}{[\mathbf{WH}]_{in}}h_{rn}}{\sum_{p=1}^{N}h_{rp}}, \tag{1.9}$$

$$w_{ir} \leftarrow \frac{w_{ir}}{\sum_{i=1}^{F}w_{ir}}, \tag{1.10}$$

where v_{in} is an entry of $\mathbf{V}$ and $r = 1, \ldots, R$. After each update of the entries in $\mathbf{W}$, the columns of $\mathbf{W}$ are normalised by means of Eq. 1.10 in order to prevent arbitrary scaling of $\mathbf{W}$ and $\mathbf{H}$. Convergence is guaranteed to a local optimum.

The innerdimension of R cannot be chosen arbitrarily. R is chosen such that $R \geq L$ with L the number of slot values in the data. There should be at least L columns in $\mathbf{W}$ (or rows in $\mathbf{H}$) to ensure a sufficient degree of freedom to prevent that crucial semantic information get lost in the factorization.

The occurence of slotvalues is correlated. Command examples in a home automation setting might contain the values <turn on> and <TV>. These two values are more likely to co-occur than <turn on> and <door>. As a result, $\mathbf{W}_s$ might integrate this correlated slot value occurrence. This would impede the learning of segregated acoustic representations for each separate slot value. To overcome this issue, the first L rows in $\mathbf{H}$ are initialized as $\mathbf{V}_s$ and the first $L \times L$ entries in $\mathbf{W}_s$ are initialized as the identity matrix [38]. A small random number is added to $\mathbf{W}_s$ and $\mathbf{H}$. The initialization procedure helps convergence to a solution with segregated slot value representations in the first L columns of $\mathbf{W}$. All entries in $\mathbf{W}_a$ are randomly initialized. We refer to the respective chapters for further initialization details.

Additional columns are usually added to $\mathbf{W}$ to catch recurring patterns that have no strict one-to-one co-occurring relation with semantic slot values. These additional columns are also called "garbage columns". Their trained content consists of acoustic fragments of spoken filler words such as "the" or "please". It also contains spoken parts of shared words among multiple slot values. For example, home automation commands might share the spoken word "door" among the expressions referring to the slot values <kitchen door> and <bathroom door>. As a consequence, the acoustic features of the spoken word "door" are not diagnostic and could as well end up in the garbage columns.

Recognition

Lets denote the trained matrices composing $\mathbf{W}$ by $\mathbf{W}_s^*$ and $\mathbf{W}_a^*$. Unlike the train set, the test set has no semantic labels. The acoustic representations of the unseen spoken test utterances is put side-by-side in the data matrix $\mathbf{V}_t$. The activations of the $\mathbf{W}_a^*$ columns is decoded by minimizing the generalized Kullback-Leibler divergence between $\mathbf{V}_t$ and $(\mathbf{W}_a^*\mathbf{H}_t)$. These activations, denoted by $\mathbf{H}_t^*$, are obtained as follows,

$$\mathbf{H}_t^* = \arg\min_{\mathbf{H}_t} D_{KL}(\mathbf{V}_t || \mathbf{W}_a^* \mathbf{H}_t) \tag{1.11}$$

The optimization problem in Eq. 1.11 is a convex problem as $\mathbf{W}_a^*$ is held fixed. The solution $\mathbf{H}_t^*$ is a global optimum and used to obtain slot value activations $\mathbf{A}$, as follows,

$$\mathbf{A} = \mathbf{W}_s^* \mathbf{H}_t^* \tag{1.12}$$

The higher the score in the rows of $\mathbf{A}$, the more likely that the respective slot value in $\mathbf{W}_a^*$ appeared in the spoken test utterance.

Note that the last step in Eq. 1.12 allows the freedom to obtain slot value activations from different columns in $\mathbf{W}_a^*$. If a semantic slot value is spread out over multiple columns in the training phase, these columns are recombined again in Eq. 1.12 and compose simular activation patterns that were labeling the spoken utterances in the training set, $\mathbf{T}$. The explained NMF procedure applies to all subsequent chapters.

1.6 The choice of the categorization method

1.6.1 *K*-means and *k*-nearest neighbours

NMF finds applications in clustering. It is demonstrated in [39] that NMF is equivalent to k-means clustering if the following conditions are met. First, the columns in $\mathbf{W}$ are considered as clustering centres. Thus each sample in $\mathbf{V}$ corresponds to one activated column in $\mathbf{W}$. Second, the objective function to minimize is the following Frobenius norm,

$$(\mathbf{W}^*, \mathbf{H}^*) = \arg\min_{\mathbf{W},\mathbf{H}} \parallel \mathbf{V} - \mathbf{WH} \parallel_F^2 \tag{1.13}$$

The Frobenius norm can be written as,

$$\parallel \mathbf{V} - \mathbf{WH} \parallel_F^2 = \sum_{n=1}^{N} \parallel \mathbf{V}_{.n} - \sum_{r=1}^{R} \mathbf{W}_{.r} \mathbf{H}_{rn} \parallel_2^2 . \tag{1.14}$$

with $\mathbf{V}_{.n}$ the data sample and $\mathbf{W}_{.r}$ the cluster centers. Since each column in $\mathbf{V}$ corresponds to one column in $\mathbf{W}$, there is only one active column entry in $\mathbf{H}$ and Eq. 1.14 is alternatively expressed as,

$$\parallel \mathbf{V} - \mathbf{WH} \parallel_F^2 = \sum_{n=1}^{N} \sum_{r=1}^{R} \delta_{r,\mathbf{H}_{rn}} \parallel \mathbf{V}_{.n} - \mathbf{W}_{.r} \parallel_2^2, \tag{1.15}$$

This norm is equivalent to the within-cluster distance in k-means.

The columns in $\mathbf{V}$ contain utterance-based feature vectors. K-means clustering provides averages to utterance-based features and do not decompose the utterance in its constitutive parts. The K-means algorithm might work properly if whole utterances are associated with the clusters. Nevertheless, the NMF approach in section 1.5.2 is able to uncover the compositional semantic structure of the utterances. This simplifies the learning problem to a great extent. For example, the card game patience has a few hundred legal utterances composed of a vocabulary with 50 essential words or less. Learning 50 words is a less complex problem than learning a few hundred utterances. In different words, NMF extends k-means in a way that it codes multiple labels and that it employs linearity.

A non-parametric classification method is k-Nearest Neighbours (k-NN, [40]). In k-NN an unseen test object is classified by a majority vote of the most common class among its k nearest neighbours. These neighbours and their associated classes are learned in the training phase. The k-NN algorithm is simple but has a few drawbacks. For example, a more frequent class tends to dominate the prediction of the new samples. A second drawback is that the accuracy degrades quickly for noisy samples with many irrelevant, or badly scaled features. A partial solution to improve robustness is to choose k sufficiently large. As for k-means, k-NN is suited for the classification of whole utterances but less appropriate for a multi-label classification problems in which labels are shared among different utterances. A more advanced approach is multi-label k-NN [41]. This approach takes into account the distribution of the labels in the k nearest neighbour set and the prior distribution that a particular label is active. However, this procedure is sound if a sufficiently large number of training samples are seen. Again, the advantage of NMF is that it uses the linearity in the data. In other approaches, the words not relevant to the label class would be considered as noise on the sample. With NMF, the linear model takes the impact of all words on the sample into account.

1.6.2 Factorization techniques

NMF is a factorization technique. Principal Component Analysis (PCA) and Singular Value Decomposition (SVD) (see [42]) are alternative factorization methods. In SVD, a data matrix $\mathbf{V}$ is decomposed in three matrices,

$$\mathbf{V}_{F\times N} = \mathbf{P}_{F\times R}\mathbf{S}_{R\times R}\mathbf{B}^T_{N\times R} \tag{1.16}$$

with $\mathbf{S}$ a square diagonal matrix. The columns of $\mathbf{P}$ are the eigenvectors of the covariance matrix $\mathbf{V}\mathbf{V}^T$ and the columns in $\mathbf{B}$ are the eigenvectors of the

covariance matrix $\mathbf{V}^T\mathbf{V}$. This is demonstrated as follows,

$$\begin{aligned}\mathbf{V} = \mathbf{PSB}^T \Leftrightarrow \mathbf{VV}^T &= \mathbf{PSB}^T\mathbf{BS}^T\mathbf{P}^T \\ &= \mathbf{PSIS}^T\mathbf{P}^T \\ &= \mathbf{PSS}^T\mathbf{P}^T \\ &= \mathbf{PS}^2\mathbf{P}^T\end{aligned}$$

$\mathbf{S}^2$ is a diagonal matrix and contains the square product of the diagonal entries in $\mathbf{S}$. Multiplication of both sides with $\mathbf{P}$ leads to the eigenvalue decomposition of the covariance matrix in PCA.

$$\begin{aligned}\mathbf{VV}^T = \mathbf{PS}^2\mathbf{P}^T \Leftrightarrow \mathbf{VV}^T\mathbf{P} &= \mathbf{PS}^2\mathbf{P}^T\mathbf{P} \\ \mathbf{VV}^T\mathbf{P} &= \mathbf{PS}^2\mathbf{I} \\ \mathbf{VV}^T\mathbf{P} &= \mathbf{PS}^2\end{aligned}$$

Obviously, the column vectors in $\mathbf{P}$ are eigenvectors of $\mathbf{VV}^T$ since $\mathbf{VV}^T\mathbf{P}$ are scalar multiples of $\mathbf{P}$. The same logic steps can be pursued for matrix $\mathbf{B}$. This demonstration shows that PCA and SVD are closely related. Since $\mathbf{P}$ composes eigenvectors, the column vectors in $\mathbf{P}$ are orthogonal and have directions with negative coordinates.

The geometric interpretation of NMF and PCA is shown in Figure 1.4 (see also [43]). The dots represent the training utterances. These are all located in the positive (hyper-)quadrant. PCA finds the main orthogonal directions in the deviation of the data from the centre, denoted by $\mathbf{P_1}$ and $\mathbf{P_2}$. By contrast, NMF constructs a convex cone with two non-negative bases $\mathbf{W_1}$ and $\mathbf{W_2}$. The convex cone represents the data by a linear combination $a\mathbf{W_1} + b\mathbf{W_2}$ and $a, b > 0$. Since the slot values and the acoustic features are represented by positive real numbers, the principal components in PCA are not a good match for the slot value composition of the data structure. By contrast, NMF provides a straightforward interpretation in which the components correspond with the compositional units that compose the whole utterance and the weights are non-negative (see Figure 1.2). Since the combination of slot values and acoustic features is located in the convex cone, the base vectors are the recurrent patterns that span this space of linear combinations. In different words, NMF allows to search acoustic-semantic co-occurring units and does not require word boundaries to do so.

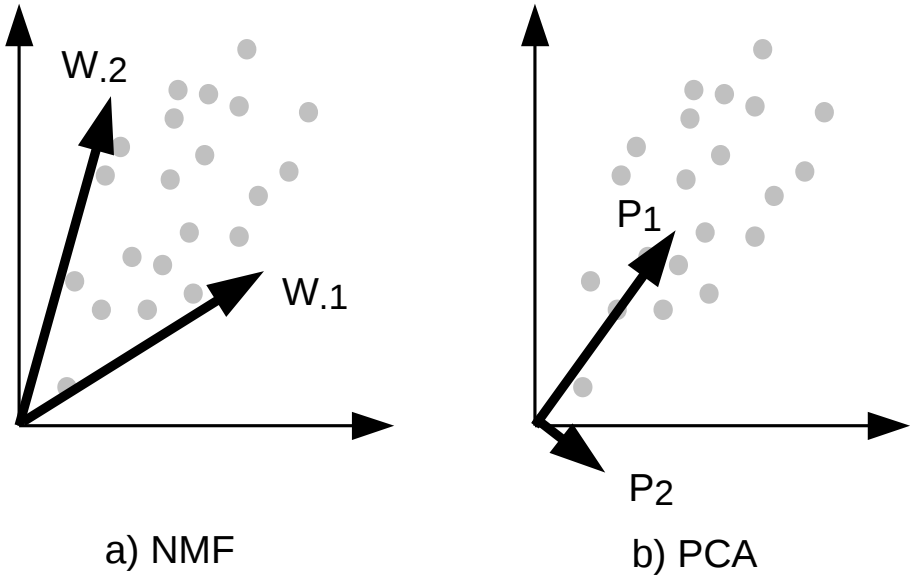


Figure 1.4: *Geometric illustration of NMF in panel a) and PCA in panel b). NMF tries to find a solution such that the data can be presented by a positive combination of basis vectors. PCA draws an orthogonal basis that explains the variance in the data.*

1.6.3 Transformation methods for multi-label classification problems

The most common transformation method for dealing with multi-label classification problems is to learn L binary classifiers, one for each different label or slot value. For each binary classifier, the original data set is partitioned into two sets. One set contains the original utterances that include the label; the second set contains the original utterances that does not contain the label. Subsequently, a binary classifier is trained for each label. In chapter 5, this transformation method precedes the training of the Gaussian Mixture Models (GMMs). Problematic is that utterances contain spoken words that are deemed to be irrelevant with respect to a single label. For each binary label classifier, these included words are added noise to the words that refer to the label.

Another problem transformation method involves the replacement of each occurring slot value ensemble by one sole label. This transforms the multi-label categorization problem into a multi-class categorization problem in which each

slot value ensemble corresponds to one spoken utterance. For some command-and-control applications, the set of all possible commands is limited and a multi-class classifier with one class per command is a feasible solution. However, this approach does not allow for generalization which is intrinsic to a semantic frame representation with multiple slots that can assume values independently. For instance, if we have learned the concept "open" and "close" from examples involving the front door, we do not need to learn these words again for opening the bathroom door.

This multi-label into a multi-class transformation approach broadens the range of possible methods and facilitates the use of typical procedures in the field of speech recognition. In typical speech recognition approaches, the choice of the input features and the categorization-regression models are mutually selected. In [44], a thorough evaluation of five different multi-layered procedures was carried out in the context of the ALADIN project. The first procedure was Dynamic Time Warping (DTW) in combination with Mel frequency cepstral coefficients (MFCC) [45]. DTW (see also [46]) is a method in which the unlabelled speech signal is compared with a large collection of labelled speech templates obtained from the training data. DTW first finds an optimal alignment between each pair of utterances and label the unlabelled speech signal with the label of the most similar template. The second procedure in [44] consisted of GMMs with 10 mixture components in conjunction with mutual-information-discriminant-analysis features (MIDA-features, see [47]). GMMs were estimated by using the expectation-maximization algorithm [48]. In the third procedure, MIDA features were employed together with three-state HMMs. The HMM is more appropriate than the GMM to model the temporal structure between subsequent frames. However, HMMs are more susceptible to overfitting because they require more parameters. All three procedures used frame-based spectrographic features. Approach four and five applied utterance-based feature vectors. One of these procedures combined a support vector machine (SVM) with GMM-supervector input ([49]). The supervectors were created in three steps. In the first step, a GMM with 512 components was trained on more than 30,000 recordings sourced from different databases. In the second step, the GMM was adapted to the speech signal of the spoken utterance. In the third step, the adapted means were concatenated in one meta feature vector. These super vectors were used to train a SVM ([50]) binary classifier for each pair of commands. A SVM is a linear classifier which constructs a hyperplane that maximizes the margins between the plane and the two classes of training examples. SVMs are also suited to perform non-linear classification by using kernels. The fifth procedure in [44] corresponds to the NMF approach (see section 1.5). Differently from our adopted approach, the classifiers in [44] differentiated between entire commands instead of separate semantic slot values. The evaluation showed that with a small amount of data, NMF outperformed the other approaches in a

rather compelling way. The second best procedure was DTW. Therefore, this procedure was also adopted in **chapter 6**.

1.6.4 Combining HAC features and NMF

The findings in [44] favoured the combination of NMF using HAC-features. NMF is a linear model that factorizes the data in meaningful chunks if data representation obeys the additivity property. Clearly, the semantic representation obeys this property because the added semantic slot values in an utterance compose the semantic representation of the utterance. HAC representations obey the additivity property if the HACs of separate words in the utterance sum to the HAC representation of the whole utterance. In Figure 1.3, the HAC diagrams of four word compositions are depicted. The upper panels contain the summed HAC diagram of the same two words, whereas the lower panels contain the HAC diagrams of the composed utterances. The words in the left and the right panel have a different chronological order. All resulting HAC diagrams are very similar except for a few counts that arise from the segmental boundaries or from speaking rate. The influence of the segmental boundaries is limited since words are composed of 30 to 200 frames on average. Since HAC features behave approximately additive, the approach of using HAC features in combination with NMF is well-suited to learn the additive acoustic representations of the additive semantic slot values.

1.7 A philosophical note on learning

The training process leads to a primitive form of understanding from a raw speech signal. This process of learning and acquiring speech understanding is at least as important as the trained models. We need to consider how meaning and understanding emerge from the operational context. This requires a different philosophy on the subject of ASR than the original ASR objectives in the Bell laboratories.

A similar renewal in thinking is noticed in many fields of cognitive science by many studies [51–58]. Cognitive science is an interdisciplinary discipline including psychology, artificial intelligence, philosophy, neuroscience and linguistics. It spans many levels of analysis, from low-level perception to high-level logic and planning. There are two dominant streams of thought on the subject matter of cognition. The oldest and most prevalent one is cognitivism [59, 60]. This approach is based on symbolic information processing. It has received a lot of attention and led to powerful representational systems.

However, in the past decades, more attention is drawn to the origin of cognition, to the organism or agent that acquires meaning by the occurrence of events and actions that affect the organism and its habitat. This approach is known as emergentism [52, 53, 55] and it influences cognitive science in a way that the focus on the design of a cognitive system is shifted to a focus on development.

Cognitivism asserts that cognition involves symbolic computations. In this process, percepts are abstracted and trigger the appropriate symbols which are merged through logic and computation, ultimately leading to an act. Sensory data provides information of the states in an objective external world. Representations of an external world form the basis of concepts that sporadically translate into language. A human designer is believed to be able to embed cognition in a system in a direct way by associating isomorphic descriptions of the objects in the external world to internal symbols. Cognition is also embedded indirectly, through a training phase in which many similar isomorphic examples are abstracted and statistically related to a symbol. This latter approach is common in ASR and complies with the approach of the Bell laboratories from the eighties. Although representations are probabilistic, the system designer is still required to identify the contextual constraints and the structure of the statistical models that delimit meaning. This approach works well as long as the system does not have to stray too far from the conditions considered by the designer or from the training material.

In the stream of emergentism, cognition is not built on objective and isomorphic representations of the external world, but on the ongoing learning process involving mutual interaction with the environment. This learning process leads to the self-organization of cognition. Co-determination between the organism and the external world determines what is meaningful. Perception and action are mixed in a dynamic interplay in which the co-dependency and coordination of perception and action are learned through development. The emergentist stream of thoughts avoids the use of words like ‘concept’ or ‘representation’. An important concept is ’embodiment’ advocating the importance of some kind of physical presence in the world in which the system operates. In the more soft sense, embodiment [61] indicates at least the existence of some sort of physical interface between the system and its world. Embodiment is also an important paradigm in the research field of developmental robotics [62].

We attempt to provide an answer to the challenges presented in section 1.4 by pursuing an approach that includes a great deal of typical emergentist features. The first, and most important one, is our focus on development in the operational environment. Albeit all reported experiments in this dissertation are carried out on a computer and thus not through an embodied system, we give a great deal of attention to a realistic simulation of the situated learning process in the environment of the user. Towards the end of the dissertation (see **chapter 6**),

we progress to a VUI model that learns incrementally with minor usage of prior knowledge. The VUI receives its input incrementally in its operational environment and thus learns the words from the user. Another emergentist property is that we treat the speech input and the executed actions (aimed output) in the same way (both as input of the VUI: receptive and proprioceptive) during the VUI learning process. More specifically, we convert and stack both information sources into one vector: one information source pertains to the input of the speech signal and one source pertains to the output that the VUI is learning to trigger. Thus instead of mapping input directly to output vectors, we associate input and output by means of loads on latent variables. These latent variables emerge from "experience", that is from the recurrence of patterns in the seen data in the operational contexts. For this approach, we used joint non-negative matrix factorization (NMF). In the following section, we give a short overview of NMF techniques that lay at the basis of our approach and that were developed prior to and during the ALADIN project.

1.8 Keyword finding

Another technique originating from Bell Laboratories is keyword spotting [63]. HMM-based models were created for keywords, irrelevant words and background noise. The keywords such as 'collect', 'person' or 'operator' were semantically significant in the domain of phone call routing. The goal of keyword spotting was to allow callers to speak in natural sentences rather than using rigid word-by-word requests. Based on HMM topologies, they were able to achieve a recognition accuracy of 95% for a keyword list of five keywords embedded in fluent normal speech [63]. Keyword finding is actually a primitive form of natural speech understanding.

Recurrent sound patterns are usually found at multiple time scales such as phones, syllables or words. Meaningful patterns will usually comply with the time scale of words or small phrases. Only if the recurrent patterns are grounded or associated with events or signals from other modalities, sound patterns can refer to something else and thus acquire some kind of meaning. For example, in [64], Independent Component Analysis (ICA) was used to decompose a drum (audio) and a silent video (video). Basis components were found that corresponded with different snare drums or bass drums and with movements of arms. However, when they decomposed an intermodal audio-video stream showing a hand playing a piano, they found basis components that were more meaningful, namely, components referring to the musical tones.

Whereas ICA produces filter components associated with low-level visual and auditory human filters, NMF provides components that are associated with

parts in a scene. NMF was introduced by Lee and Seung [37, 65] in a paper that described its application to face recognition. It was demonstrated that NMF can decompose an image into localised features such as an eye brow or lips from a face. By using sparsity constraints [66], even more localised features could be found. It was also used on audio magnitude spectra [67] to decompose a musical composition.

The use of NMF for the decomposition of speech into its word constituents was pursued in [68, 69]. In these studies, NMF works by factorizing a collection of utterance-based representations into the product of a matrix containing the latent factors describing the recurrent acoustic patterns and a matrix containing the incidence of these patterns. The utterance-based representations consisted of an accumulation of the probability of every consecutive phone pair in an utterance. The factorisation in [68, 69] revealed digit words as basis components from the TI-Digits corpus. The method was extended in [34] where fixed-length vector-quantized (VQ) spectral input vectors were composed and grounded with semantic information. The semantic information was represented by fixed-length vectors denoting the presence or absence of keywords. The factorization revealed latent factors corresponding to words and meaningful phrases. The method was further developed in the ACORNS project [70–72] as a computational model of vocabulary acquisition. The ACORNS corpus was composed of 50 keywords embedded in artificial sentences. In [73], the NMF technique was made adaptive to changes in grounding relations. Although all these NMF-based methods work well with prior definitions of the phonetics or the vector-quantized spectral features, this may be inadequate for acoustic representations of dysarthric speech. The wide-ranged differences in the phonetic regularities and the acoustic clusters among pathologies and speakers appeals for a more tailored solution. A better approach may be the development of personalised systems with speaker-adapted representations of acoustic word realizations.

When NMF input vectors consist of phone occurrences, NMF decomposes utterances naturally into words without the need for grounding or supervision [68, 69]. However, less structured acoustic atoms require supervision to guide the factorization towards a word-like decomposition [34]. If the acoustic input is augmented with semantic content from the action scene embedding the spoken utterance, a decomposition with latent factors referring to the semantic content will be promoted. A phone model is more informative and operates on a longer time-scale than VQ spectral clusters. By using acoustic features on a smaller time-scale, less prior knowledge is integrated beforehand and word models are built from more fine-grained structure. The advantage is that the acoustic representation is less constrained and more suitable for dysarthric speech, nevertheless, more training data might be needed to build words from the more fine-grained structure.

Different kinds of acoustic pre-processing can be used and paired with semantic content to build NMF input vectors. All these can be investigated on their adequacy to respond to the challenges presented in section 1.4. In Figure 1.5, a schematic overview depicts all sorts of processing pathways that have been investigated, before, in parallel with, and within the ALADIN project (see **chapter 3**). It is a multi-layered hierarchical and modular scheme that pertains to the development of acoustic NMF input features. The feature representations unfold from bottom to top, and the multiple arrow directions indicate various combinations of processing procedures. The architecture is modular in the sense that each procedure consist of a self-contained module, affording many interesting opportunities like replacing one method by the other at the same layer, keeping all other processing modules identical.

First, the spectro-temporal features are extracted from the speech signal and processed to either log Mel-spectral features; to MFCC features; to features obtained by mutual information discriminant analysis (MIDA, [47]) using phonetic transcriptions as target classes (MIDA phone, [74]); to MIDA features using VQ clusters as target classes (VQ-MIDA, [75]); or to modulation spectrogram (MS, [76]) features. All these features have been used in a NMF-based context, albeit not always for the aim of word learning. The horizontal red arrow leading to the MIDA features from the left indicates that for the creation of MIDA phone features, annotated speech material is needed. The horizontal green arrow from the right indicates that for the creation of VQ-MIDA features, unsupervised speech material is needed.

In three methods, the feature vector obtained by the different feature extraction methods were transformed into posteriorgrams, either by using a phone recogniser [68, 74, 77], by using self-discovered sub-word units [77, 78] or by clustering the data with VQ clusters or Gaussians [34, 73, 74, 79]. As for the phone MIDA feature extraction, the horizontal red arrow leading to the phone recogniser from the left indicates that for this approach, annotated speech material is needed, while the horizontal arrows leading to the soft VQ/Gaussian procedure and the self-discovered sub-word units from the right indicate that data-driven speech material is needed to train the code books or extract the subword units. In the mid-level layer, the acoustic input was also enhanced by using exemplars drawn from unsupervised speech material, by using self-discovering subword units, phones or clusters. The last three units are represented by posteriorgrams. The posteriorgram of an utterance has a variable length that depends on the number of frames in an utterance. However, fixed-length vectors are required to compose the data matrix for NMF. Posteriorgrams are converted to utterance-based HAC representations after which the NMF training takes place. The aim of HAC [34, 69] is to build a fixed-length vector for each utterance by accumulating the probability of

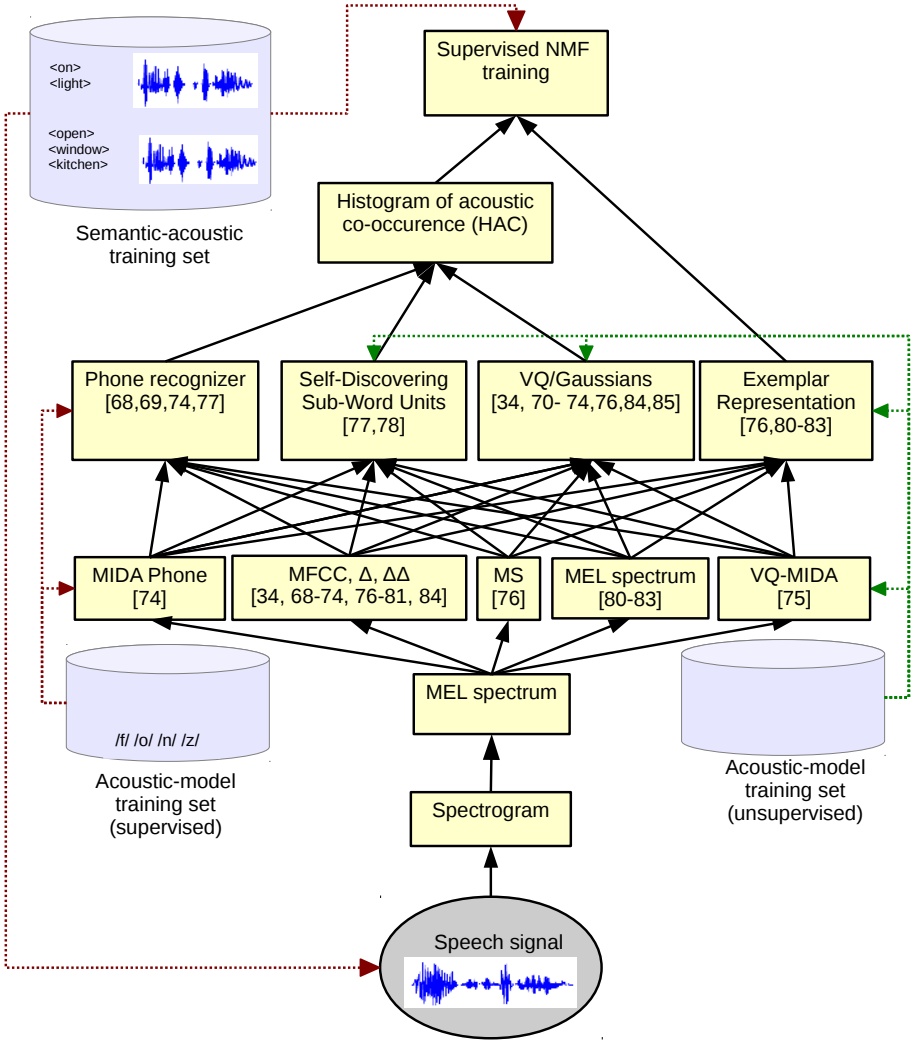


Figure 1.5: *multi-layered hierarchical and modular overview*

observing a phones or a code word pair for all possible pairs in a phone lattice or in the time domain, shifted a specified number of frames away from each other. Multiple lags or shifts can be taken into consideration incorporating more time information [75].

A fourth method that build more advanced feature representations is exemplar-based sparse representations [76, 80–83]. Utterance-based feature vectors consist of sparse vectors of coefficients describing the input signal as a linear combination of speech exemplars, i.e. selected segments of real speech, from a dictionary. Since the exemplars may have a longer duration, these representations are more robust against temporal variations in speech. Also, since time information is already encoded in these coefficients, there is no need to consider procedures like HAC representations in order to include time information in the utterance-based feature representation. In [35], parameters for the use of exemplar-based sparse representations of speech were explored like the influence of the total number of exemplars considered and the sparseness imposed on their linear combinations. The best results were obtained by imposing a strong sparsity and using many exemplars.

A criticism to NMF is that chronological information is lost. NMF detect words in an utterance but not the chronological order. NMF is sometimes extended to find time-dependent patterns. An example is time coded NMF [84], allowing the detection of patterns in time. The same problem was addressed in [85] and in [86] by combining sliding window and convolutive NMF. Graph regularisation in NMF [79] is another procedure that includes the chronological sequence in the factorization.

1.9 Outline of the thesis

The dissertation is organized in this introductory chapter, five chapters based on the author's published, accepted and submitted peer-reviewed papers and a concluding chapter. The order of the chapters follows the chronological line of research. Each chapter describes a subject matter pertaining to the goals exposed in section 1.4. Each chapter starts with a contextual note framing each piece of research in the global picture of the research project, the chapters are all self-contained; thus providing all required formalism to back up the experimental designs.

The followed approach is based on joint factorization of labels and features. The VUI infers its own labels from actions that are demonstrated by the user, thus we expect a significant proportion of erroneous learning examples. An essential condition of this learning approach is its robustness against possible

erroneous grounding. A method to improve label noise robustness in joint NMF is presented in [87]. This method leaves out the training examples when their label is uncertain. In **chapter 2**, we test the robustness of the NMF approach against grounding errors and envisaged four possible ways how grounding in an embodied vocal interface might fail. It appeared that joint NMF decomposition of acoustic and label data is very robust against substantial proportions of errors in the label data. This facilitates the learning procedure, because the dysarthric speakers do not need to correct all errors. The results also show that the issue of supervision errors did not require further investigation in the framework of the ALADIN project. We therefore decided to move on to the next research topics.

In **chapter 3**, we investigated different processing flows and represented a modular system for building up NMF input vectors. This modularity allowed us to conduct an exhaustive survey on different processing flows and their stepwise improvements. We introduced phone posteriorgrams for the purpose of fast learning and combined multiple streams. In former NMF approaches, the data was pooled over multiple speakers. We advanced to speaker-dependent models at different levels of the system. This personalization of the VUI proved to be of great importance and its success inspired us to put more effort in learning models that acquire most of their knowledge during usage. This ultimately led to the probabilistic incremental models in **chapter 6**. All conducted experiments in the chapters following **chapter 3** were using speaker-dependent NMF. In **chapter 3**, we also proposed a learning curve as a basis to evaluate learning performance. This evaluation procedure is used throughout the text and facilitate evaluation between chapters.

Whereas **chapter 3** mainly dealt with feature representations, we investigated the NMF approach in **chapter 4**. In this chapter, we propose two measures to improve learning from scarce data: smoothing of posteriorgrams and constraining the number of free parameters. Both measures improved learning speed. However, these improvements raised questions about the effectiveness of the NMF approach. It was an indication that the optimization problem is susceptible to overfitting. This problem is alleviated by pursuing Baysian procedures in **chapter 6**.

The data collection with dysarthric speech was not available at the time and for this practical reason it took a while before the research on dysarthric speech took off. The first conducted research recorded in the ALADIN project is reported in [29]. Since this research mainly focused on grammar induction, it is left out of the scope of this dissertation. In this dissertation, the first results on dysarthric speech are presented in **chapter 5**. In this chapter we evaluate learning in the VUI using two corpora, one with recordings of dysarthric spoken commands related to home automation command-and-control, and one with recordings of

normal speech with speakers playing a card game patience. We investigated the performance for different semantic frame structures and proposed an unbiased decision process to predict the semantic frame.

In **chapter 6**, we introduced incremental learning algorithms to learn from scratch. The proposed methods are memoryless and have low computational complexity. In an extensive comparative study, we compared all incremental learning algorithms with their batch variants and focused on their capacity to learn from dysarthric speech and to adapt to new vocalizations.

All chapters deal with the feasibility of the vocal user interface and the learning rate. These concerns are important to keep the motivation of users going while they are training their system. We return to this concern in the conclusion of the thesis and thoughts on future work in **chapter 7**.

1.10 References

[1] H. Van hamme, P. Deboutte, D. De Grooff, B. Vanrumste, and W. Daelemans, "Aladin, adaptation and learning for assistive domestic vocal interfaces." project proposal for the SBO call of IWT-Flanders. pages 4

[2] M. Hawley, P. Enderby, P. Green, S. Cunningham, S. Brownsell, J. Carmichael, M. Parker, A. Hatzis, P. O?Neill, and R. Palmer, "A speech-controlled environmental control system for people with severe dysarthria," *Medical Engineering & Physics*, vol. 29, no. 5, pp. 586–593, 2007. pages 5, 8

[3] R. Palmer, P. Enderby, and M. Hawley, "Addressing the needs of speakers with longstanding dysarthria: computerised and traditional therapy compared," *International Journal of Language and Communication Disorders*, vol. 42, no. 1, pp. 61–79, 2007. pages

[4] M. Parker, S. Cunningham, P. Enderby, M. Hawley, and P. Green, "Automatic speech recognition and training for severely dysarthric users of assistive technology: the stardust project," *Clinical linguistics & phonetics*, vol. 20, no. 2-3, pp. 149–156, 2006. pages

[5] H. M.S., "Speech recognition as an input to electronic assistive technolog," *British Journal of Occupational Therapy*, vol. 65, no. 1, pp. 15–20, 2002. pages 5, 8

[6] B. H. Juang and L. R. Rabiner, "Automatic speech recognition - a brief history of the technology development," 2005. pages 6

[7] B. Juang, "Maximum-likelihood estimation for mixture multivariate stochastic observations of markov chains," *AT&T technical journal*, vol. 64, no. 6, pp. 1235–1249, 1985. pages 6

[8] C. Lee, L. Rabiner, R. Pieraccini, and J. Wilpon, "Acoustic modeling for large vocabulary speech recognition," *Computer Speech & Language*, vol. 4, no. 2, pp. 127–165, 1990. pages

[9] L. R. Bahl, F. Jelinek, and R. Mercer, "A maximum likelihood approach to continuous speech recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 2, pp. 179–190, 1983. pages

[10] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An introduction to the application of the theory of probabilistic functions of a markov process to automatic speech recognition," *Bell System Technical Journal, The*, vol. 62, no. 4, pp. 1035–1074, 1983. pages 6

[11] S. Furui, "Speaker-independent isolated word recognition using dynamic features of speech spectrum," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 34, no. 1, pp. 52–59, 1986. pages 6

[12] K.-F. Lee and H.-W. Hon, "Speaker-independent phone recognition using hidden markov models," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 37, pp. 1641–1648, Nov 1989. pages 6

[13] F. Jelinek, "The development of an experimental discrete dictation recognizer," in *Informatik-Anwendungen?Trends und Perspektiven*, pp. 109–117, Springer, 1986. pages 6

[14] L. E. Baum, "An equality and associated maximization technique in statistical estimation for probabilistic functions of markov processes," *Inequalities*, vol. 3, pp. 1–8, 1972. pages 6

[15] G. Hinton, L. Deng, D. Yu, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. S. G. Dahl, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, pp. 82–97, November 2012. pages 6

[16] J. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *Speech and Audio Processing, IEEE Transactions on*, vol. 2, pp. 291–298, Apr 1994. pages 7

[17] C. Leggetter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer Speech & Language*, vol. 9, no. 2, pp. 171 – 185, 1995. pages 7

[18] R. Kuhn, P. Nguyen, J.-C. Junqua, L. Goldwasser, N. Niedzielski, S. Fincke, K. Field, and M. Contolini, "Eigenvoices for speaker adaptation.," in *ICSLP*, vol. 98, pp. 1774–1777, 1998. pages 7

[19] T. Macrae, A. A. Tyler, and K. E. Lewis, "Lexical and phonological variability in preschool children with speech sound disorder," *American Journal of Speech-Language Pathology*, vol. 23, no. 1, pp. 27–35, 2014. pages 7

[20] H. Christensen, I. Casanuevo, S. Cunningham, P. Green, and T. Hain, "homeservice: Voice-enabled assistive technology in the home using cloud-based automatic speech recognition," *Proc SLPAT*, pp. 29–34, 2013. pages 8

[21] J. Derboven, J. Huyghe, and D. De Grooff, "Designing voice interaction for people with physical and speech impairments," in *Proc. NordiCHI*, (Helsinki, Finland), pp. 217–226, 2014, October. pages 9

[22] J. Huyghe, J. Derboven, and D. De Grooff, "Aladin: Adaptive voice interface for people with disabilities," in *proc NordiCHI, Workshop on: "Designing Self-care for Everyday Life"*, (Helsinki, Finland), 2014, October. pages

[23] J. Huyghe, J. Derboven, and D. De Grooff, "Aladin: demo of a multimodal adaptive voice interface," in *Proc. NordiCHI*, (Helsinki, Finland), pp. 1035–1038, 2014, October. pages

[24] J. Huyghe, J. Derboven, and D. Geerts, "Aladin: Adaptive speech interaction for people with disabilities," (Toronto, Canada), 2014, April. pages 9

[25] J. Van de Loo, G. De Pauw, J. F. Gemmeke, P. Karsmakers, B. Van Den Broeck, W. Daelemans, and H. Van hamme, "Grammar induction for assistive domestic vocal interfaces," *the 22nd Meeting of Computational Linguistics in the Netherlands (CLIN22), Tilburg, The Netherlands*, 20/01/2012 2012. pages 9

[26] *A Self-Learning Assistive Vocal Interface Based on Vocabulary Learning and Grammar Induction*, (Portland, Oregon, USA), 09/2012 2012. pages

[27] J. Van de Loo, J. F. Gemmeke, G. De Pauw, W. Daelemans, and H. Van Damme, "Weakly supervised semantic frame induction: effects of using background knowledge," *Presented at the 24th Meeting of Computational Linguistics in the Netherlands (CLIN 2014), Leiden, The Netherlands*, 17/01/2014 2014. pages

[28] J. Van de Loo, G. De Pauw, J. F. Gemmeke, and W. Daelemans, "Weakly supervised concept tagging: combining a generative and a discriminative approach," *Presented at the 25th Meeting of Computational Linguistics in the Netherlands (CLIN 2015), Antwerp, Belgium*, 06/02/2015 2015. pages 9

[29] B. Ons, N. Tessema, J. Van De Loo, and J. F. Gemmeke, "A self learning vocal interface for speech-impaired users," in *Proceedings SLPAT 2013*, pp. 1–9, 2013. pages 9, 32

[30] G. Dekkers, T. van Waterschoot, B. Vanrumste, B. Van Den Broeck, J. F. Gemmeke, H. Van hamme, and P. Karsmakers, "A multi-channel speech enhancement framework for robust NMF-based speech recognition for speech-impaired users," 2015. pages 9

[31] H. H. Clark and E. F. Schaefer, "Contributing to discourse," *Cognitive science*, vol. 13, no. 2, pp. 259–294, 1989. pages 11

[32] C. Middag, *Automatic Analysis of Pathological Speech*. PhD thesis, Ghent University, Belgium, 2012. pages xiii, 12

[33] Y. Wang and A. Acero, "Rapid development of spoken language understanding grammars," *Speech Communication*, vol. 48, no. 3-4, pp. 390–416, 2006. pages 13

[34] H. Van hamme, "HAC-models: a novel approach to continuous speech recognition," in *Proc. Interspeech*, (Brisbane, Australia), pp. 255–258, 2008. pages 16, 28, 29

[35] J. Driesen, J. Gemmeke, and H. Van hamme, "Weakly supervised keyword learning using sparse representations of speech," in *Proc. ICASSP*, (Kyoto, Japan), pp. 5145–5148, 2012. pages 18, 31

[36] C. A., Z. R., P. A.H., and A. S-I., *Nonnegative Matrix and Tensor Factorizations Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. John Wiley & Sons, Ltd, 2009. pages 18

[37] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, pp. 556–562, 2001. pages 18, 28

[38] J. Driesen, *Discovering words in speech using matrix factorization*. PhD thesis, K.U.Leuven, ESAT, July 2012. pages 19

[39] C. H. Ding, X. He, and H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering.," in *SDM*, vol. 5, pp. 606–610, SIAM, 2005. pages 20

[40] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. IT-13, no. 1, pp. 21–27, 1967. pages 21

[41] M.-L. Zhang and Z.-H. Zhou, "Ml-knn: A lazy learning approach to multi-label learning," *Pattern recognition*, vol. 40, no. 7, pp. 2038–2048, 2007. pages 21

[42] D. Lay, *Linear algebra and its applications*. Pearson, 2012. pages 21

[43] M. Sun, *Constrained non-negative matrix factorization for vocabulary acquisition from continuous speech*. PhD thesis, K.U.Leuven, ESAT, July 2012. pages 22

[44] L. Broekx, K. Dreesen, J. F. Gemmeke, and H. Van hamme, "Comparing and combining classifiers for self-taught vocal interfaces," in *Proc SLPAT*, (Grenoble, France), pp. 21–28, 2013. pages 24, 25

[45] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *In IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980. pages 24

[46] G. Aimetti and R. K. Moore, "A computational model of preverbal infant word learning," in *International Conference on Cognitive Modeling*, 2009. pages 24

[47] K. Demuynck, *Extracting, Modelling and Combining Information in Speech Recognition*. PhD thesis, K.U.Leuven, ESAT, February 2001. pages 24, 29

[48] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–38, 1977. pages 24

[49] M. H. Bahari and H. Van hamme, "Speaker age estimation using hidden markov model weight supervectors," in *International Conference on Information Science, Signal Processing and their Applications (ISSPA)*, pp. 517–521, IEEE, 2012. pages 24

[50] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995. pages 24

[51] A. Clark, *Mindware: An Introduction to the Philosophy of Cognitive Science*. New York: Oxford University Press, 2001. pages 25

[52] H.-H. Nagel, "Steps toward a cognitive vision system," *AI magazine*, vol. 25, no. 2, p. 31, 2004. pages 26

[53] M. Wilson, "Six views of embodied cognition," *Psychonomic bulletin & review*, vol. 9, no. 4, pp. 625–636, 2002. pages 26

[54] C. Fernando and S. Sojakka, "Pattern recognition in a bucket," in *Advances in Artificial Life* (W. Banzhaf, J. Ziegler, T. Christaller, P. Dittrich, and J. Kim, eds.), vol. 2801 of *Lecture Notes in Computer Science*, pp. 588–597, Springer Berlin Heidelberg, 2003. pages

[55] D. Vernon, G. Metta, and G. Sandini, "A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents," *IEEE Transactions on Evolutionary Computation*, vol. 11, pp. 151–180, April 2007. pages 26

[56] E. D. Dickmanns, "Dynamic vision-based intelligence," *AI Magazine*, vol. 25, no. 2, p. 10, 2004. pages

[57] J. P. Crutchfield, "Dynamical embodiments of computation in cognitive processes," *Behavioral and Brain Sciences*, vol. 21, pp. 635–635, 10 1998. pages

[58] C. Town and D. Sinclair, "A self-referential perceptual inference framework for video interpretation," in *Computer Vision Systems* (J. Crowley, J. Piater, M. Vincze, and L. Paletta, eds.), vol. 2626 of *Lecture Notes in Computer Science*, pp. 54–67, Springer Berlin Heidelberg, 2003. pages 25

[59] R. M. Gagne, "The conditions of learning..," 1970. pages 25

[60] J. M. Scandura, *Structural learning: theory and research*, vol. 1. Gordon and Breach, 1973. pages 25

[61] T. Ziemke, "What's that thing called embodiment," in *Proceedings of the 25th Annual meeting of the Cognitive Science Society*, pp. 1305–1310, Mahwah, NJ: Lawrence Erlbaum, 2003. pages 26

[62] M. Asada, K. F. MacDorman, H. Ishiguro, and Y. Kuniyoshi, "Cognitive developmental robotics as a new paradigm for the design of humanoid robots," *Robotics and Autonomous Systems*, vol. 37, no. 2–3, pp. 185 – 193, 2001. Humanoid Robots. pages 26

[63] J. Wilpon, C. Lee, and L. Rabiner, "Application of hidden markov models for recognition of a limited set of words in unconstrained speech," in *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*, pp. 254–257 vol.1, May 1989. pages 27

[64] P. Smaragdis and M. Casey, "Audio/visual independent components," in *Proc. ICA*, pp. 709–714, 2003. pages 27

[65] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999. pages 28

[66] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *The Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004. pages 28

[67] P. Smaragdis and J. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.*, pp. 177–180, Oct 2003. pages 28

[68] V. Stouten, K. Demuynck, and H. Van hamme, "Automatically learning the units of speech by non-negative matrix factorisation," in *Proc. European Conference on Speech Communication and Technology*, pp. 1937–1940, 2007. pages 28, 29

[69] V. Stouten, K. Demuynck, and H. Van hamme, "Discovering phone patterns in spoken utterances by non-negative matrix factorization," *Signal Processing Letters, IEEE*, vol. 15, pp. 131–134, 2008. pages 28, 29

[70] J. Driesen and H. Van hamme, "Modelling vocabulary acquisition, adaptation, and generalization in infants using adaptive bayesian plsa," *Neurocomputing*, vol. 74, pp. 1874–1882, 2011. pages 28

[71] L. ten Bosch, J. Driesen, H. Van hamme, and L. Boves, "On a computational model for language acquisition: modeling cross-speaker generalisation," in *Text, Speech and Dialogue*, pp. 315–322, Springer, 2009. pages

[72] L. ten Bosch, H. Van hamme, and L. Boves, "A computational model of language acquisition: focus on word discovery," in *Proc Interspeech 2008*, Citeseer, 2008. pages 28

[73] J. Driesen, L. ten Bosch, and H. Van hamme, "Adaptive non-negative matrix factorization in a computational model of language acquisition," in *Proc. Interspeech*, (Brighton, UK), pp. 1711–1714, 2009. pages 28, 29

[74] B. Ons, J. F. Gemmeke, and H. Van hamme, "Fast vocabulary acquisition in an NMF-based self-learning vocal user interface," *Computer Speech & Language*, vol. 28, no. 4, pp. 997 – 1017, 2014. pages 29

[75] J. Driesen, J. F. Gemmeke, and H. Van hamme, "Data-driven speech representations for NMF-based word learning," in *Proc. of the workshop on Statistical and Perceptual Audition*, (Portland, OR, USA), 2012. pages 29, 31

[76] D. Baby, T. Virtanen, J. Gemmeke, T. Barker, and H. Van hamme, "Exemplar-based noise robust automatic speech recognition using modulation spectrogram features," in *Proc. IEEE Spoken Language Technology Workshop*, pp. 1–6, 2014. pages 29, 31

[77] M. Sun and H. Van hamme, "Joint training of non-negative tucker decomposition and discrete density hidden markov models," *Computer Speech & Language*, vol. 27, no. 4, pp. 969 – 988, 2013. pages 29

[78] J. Driesen and H. Van hamme, "Fast word acquisition in an NMF-based learning framework," in *Proc. ICASSP*, (Kyoto, Japan), pp. 5137–5140, 2012. pages 29

[79] M. Sun and H. Van hamme, "Unsupervised vocabulary discovery using non-negative matrix factorization with graph regularization," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 5152–5155, May 2011. pages 29, 31

[80] J. Gemmeke and H. Van hamme, "Advances in noise robust digit recognition using hybrid exemplar-based techniques," *Proceedings Interspeech 2012*, pp. 1–4, 2012. pages 31

[81] J. F. Gemmeke, T. Virtanen, and A. Hurmalainen, "Exemplar-based sparse representations for noise robust automatic speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 7, pp. 2067–2080, 2011. pages

[82] J. F. Gemmeke, T. Virtanen, and A. Hurmalainen, "Exemplar-based speech enhancement and its application to noise-robust automatic speech recognition," in *Proc. 1st Int. Workshop on Machine Listening in Multisource Environments (CHiME)*, pp. 53–57, 2011. pages

[83] E. Yílmaz, J. F. Gemmeke, and H. Van hamme, "Noise robust exemplar matching using sparse representations of speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 22, pp. 1306–1319, Aug. 2014. pages 31

[84] H. Van hamme, "An on-line NMF model for temporal pattern learning: Theory with application to automatic speech recognition," in *Latent Variable Analysis and Signal Separation* (F. Theis, A. Cichocki, A. Yeredor, and M. Zibulevsky, eds.), vol. 7191 of *Lecture Notes in Computer Science*, pp. 306–313, Springer Berlin Heidelberg, 2012. pages 31

[85] W. Cotteleer, "The self-learning vocal interface," Master's thesis, K.U.Leuven, ESAT, 2012. pages 31

[86] P. Smaragdis, "Convolutive speech bases and their application to supervised speech separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, pp. 1–12, Jan 2007. pages 31

[87] M. Versteegh, L. ten Bosch, and L. Boves, "Active word learning under uncertain input conditions," in *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association*, (Makuhari, Chiba, Japan,), pp. 2930–2933, 2010. pages 32

Chapter 2

Lable noise robustness

This chapter is based on the following article:
B., Ons, J.F., Gemmeke, and H., Van hamme, "Label noise robustness and learning speed in a self-learning vocal user interface," *in Proc. of the international workshop on spoken dialog systems (IWSDS)*, (Ermenonville, France),2012.

2.1 Abstract

A self-learning vocal user interface (VUI) learns to map user-defined spoken commands to intended actions. The VUI is trained by mining the speech input and the demonstrated action on a device. Although this generic procedure allows a great deal of flexibility, it comes at a cost. Two requirements are important to create a user-friendly learning environment. First, the self-learning interface should be robust against typical errors that occur in the interaction between a non-expert user and the system. For instance, the user gives a wrong learning example to the system by commanding "Turn on the television" and pushing the power button of the digibox. The spoken command is then supervised by a wrong action and we refer to these errors as label noise. Secondly, the mapping between voice commands and intended actions should only require a few examples. To meet these requirements, we implemented learning through supervised non-negative matrix factorization (NMF). We tested keyword recognition accuracy for different levels of label noise and different sizes of training sets. Our learning approach is robust against label noise but some improvement regarding fast mapping is desirable.

2.2 Context and contributions of the chapter

In [1], NMF was introduced as a machine learning algorithm for keyword learning and keyword spotting. The speech signal of spoken utterances was accumulated in utterance-based feature vectors and augmented with a vector indicating keyword occurrence. Although the exact keyword time stamps were not required, the method was successful in separating acoustic keyword features and achieving viable keyword spotting results.

The idea in the ALADIN project was to build a VUI that learns commands from the user in a similar way as the former keyword spotting technique. Keyword occurrence vectors were traded for vectors that indicate the occurrence of words or phrases referring to semantic content in spoken commands. Semantic description vectors could be obtained from the exchange of information between the VUI and the connected applications. The former keyword spotting procedure was adopted as a computational model of vocabulary acquisition [2], and here the approach is developed into the conception of a vocal user interface (VUI).

The self-learning aspect of the vocal user interface refers to the training of the VUI by the (non-expert) user. The emergentist theories of language acquisition (see chapter 1) is a source of inspiration to reflect on the user, his habitat and how the user trains the VUI. The user is part of the environment and

becomes an interactive agent during training. As a consequence, VUI learning depends on data mining of the correctly executed demonstrations by non-expert users. The first contribution of the chapter is the realization that situated learning with non-expert users might become problematic. It is important to investigate the robustness of the NMF approach against plausible mistakes that emerge during training by non-expert users. The second contribution is the consideration and formalization of the different kinds of label noise that could emerge in the situated learning context. A third contribution that was not mentioned in the published version of the manuscript was the introduction of multiple streams based on codebooks with different codebook sizes. Different streams based on different code books have been used before, but here, we introduce different codebook sizes with the aim of representing information on different scales of granularity in the acoustic feature space.

The data collection targeting the ALADIN VUI approach was not available at the time of writing the manuscript. Therefore, we used the ACORNS corpus [13] instead. The advantage of using this corpus was the possibility to compare the performance with foregoing conducted experiments in [1, 3]. The baseline for evaluating further improvements was established.

2.3 Introduction

In *Command-and-control* speech recognition, the user usually speaks a phrase from a set of predefined grammars and words. Large transcribed speech data sets are used to train the acoustic model beforehand and the language model is often written by hand in the form of context-free grammars. Such models fit well for users that a developer in a lab had in mind. Although these models suit the average speaker very well, they are inadequate for speakers with deviant speech. There is a renewed interest in dialogue systems that allow for more freedom in the interaction between man and machine by means of adaptation ([4], [5]). However, these systems are still based on predefined language and acoustic models that adapt through interactions to real-life situations.

Contrarily, we aim to design a vocal user interface that learns to understand normal or deviant speech by associating the spoken commands and their related actions during its usage (see [6]). The vocal user interface is trained by the end user by mining the speech input and the changes that are provoked on the device. The end user is able to specify his own commands and trains the system by giving examples to the system. For instance, the user might say: "Please, turn on the television" and turns on the television with the remote control. The learning problem is a machine learning problem where the user has to demonstrate the intended action to the vocal user interface, and by

doing so, he provides supervision to the machine learning process. The vocal user interface should learn the association between the vocal command and the intended action and it should control the intended action in future command calls by the user.

In a command-and-control speech application, some words are meaningful while others are not. For instance, for the command: "Please, turn on the television", the informative parts are "turn on" and "television", but the polite introduction "please" is not informative. The required information needed to control a device is solely determined by the device. Informative parts lead to the identification of the desired action and we call these parts "keywords", while the others are referred to as "filler words".

When we assign a label to each keyword, the learning problem can be redefined: the vocal user interface should learn the association between the spoken keywords and the labels associated with the appropriate action. Because supervision is solely depending on the examples provided by the end user, correct demonstrations of consistent spoken commands will be more effective for learning all necessary associations between spoken keywords and labels. However, instead of imposing consistency on the user, we would rather prefer to design a user-friendly system where natural variation in communications with the system is allowed. For instance, when the user stands in front of the television, the user might say "Turn on, please!" and turns on the television. The provoked action is associated with the keyword labels "turn on" and "television" while the last keyword is actually missing in the acoustic input. Another kind of error would occur if the user says, "Turn on the television" but mistakenly pushes the power button on the remote control of the audio system. Obviously, such errors occur easily in the natural interaction between man and machine, and the learning algorithm should be robust against them. We refer to these errors with the term *label noise* (see [7] for a discussion of the impact of label noise in clustering methods).

Since the end-user has to provide some effort to train the device, it is desirable that mappings should be learned fast. Label noise robustness and fast learning are two requirements that we investigate in the current study. In fact, the two may be related because the speed of learning might slow down when the system is not robust against it. Conversely, the system might be less robust against label noise when the system is able to learn mappings from only a few examples as the number of examples might be too low to obtain a representative sample for future data.

We have chosen a supervised non-negative matrix factorization (NMF) approach [1, 3, 8–10] to learn the mappings between the spoken keywords and the keyword labels. NMF is a method to learn the underlying patterns in data like

speech, images [11], documents [12] and many other types of data. Supervised NMF allows to discover latent patterns in data signalling the occurrence of meaningful parts like for instance the keywords in spoken commands and dialogues. The strength of the NMF approach is that the acoustic representation of a keyword can be found through weak supervision. We only have to specify which keywords are spoken in which utterance. NMF is able to extract the acoustic representation of a keyword from the examples of multiple weakly supervised spoken utterances. The goal of the current study is to test the two previously mentioned requirements, i.e label noise robustness and fast learning, within the context of a supervised NMF framework.

The rest of the chapter is organized as follows. In Section 2.4, we briefly explain supervised NMF learning. In Section 2.5, we list four different types of errors that we expect to occur in the user's environment and we explain the effect of these errors for the NMF framework. In Section 2.6, we discuss the experimental design and we provide all technical details that are specific to the current experiments. Finally, in Section 2.7 and Section 2.8 we show and discuss the results.

2.4 Supervised word learning

NMF is a machine learning approach aiming at the discovery of latent structure in data based on the decomposition of a larger data matrix $\mathbf{V}$ into the product of two matrices $\mathbf{W}$ and $\mathbf{H}$ of lower dimensionality.

$$\mathbf{V} \approx \mathbf{WH} \tag{2.1}$$

The colums in $\mathbf{W}$ represent the latent structure (recurring patterns) of the colums in $\mathbf{V}$ and the colums in $\mathbf{H}$ indicate which patterns are combined to approximate the colums in $\mathbf{V}$. Iterative update rules for minimizing a distance measure between $\mathbf{V}$ and ($\mathbf{WH}$) can be found in [1, 10, 11]. In the current NMF keyword learning approach, we have chosen the Kullback-Leibler divergence as the distance measure.

In our approach, the matrices $\mathbf{V}$, $\mathbf{W}$ and $\mathbf{H}$ can be interpreted as follows. The n^{th} utterance in the training set is represented by the column $\mathbf{v}_n$ in $\mathbf{V}$ ($n = 1 \ldots N$). Thus, the colums in $\mathbf{V}$ comprise the learning examples (vectorized in a column vector) provided by the user. Supervised NMF learning leads to columns in $\mathbf{W}$ corresponding to keywords while columns in $\mathbf{H}$ indicate which columns in $\mathbf{W}$ are combined to compose the keywords in the utterance of the spoken command $\mathbf{v}_n$.

In supervised NMF learning, (see [1] and [10]), the observation data of the n^{th} utterance $\mathbf{v}_n$ in the learning phase consists actually of two parts: the acoustic representation of the command spoken by the user and the keyword labels in the action handled by the machine. We denote the acoustic part of $\mathbf{V}$ by $\mathbf{V}_a$ and the part indicating the presence of keywords by $\mathbf{V}_l$. For each to-be-learned keyword label, there is one row foreseen in $\mathbf{V}_l$ and its entries represent the number of times that the respective keyword was uttered in the n^{th} utterance. In the matrix $\mathbf{W}$, an equal number of rows for the keyword labelling part are added to the acoustic part of $\mathbf{W}$. Keyword labels in $\mathbf{W}$ consist of 1 on row k for the k^{th} keyword and 0 elsewhere. The purpose of the supervised NMF learning is to find the latent acoustic representations of the keywords (the acoustic part of $\mathbf{W}$). When we denote the labelling part of $\mathbf{W}$ by $\mathbf{W}_l$ and the acoustic part by $\mathbf{W}_a$, Equation 2.1 (see [1] and [10]) is extended to

$$\begin{bmatrix} \mathbf{V}_l \\ \\ \mathbf{V}_a \end{bmatrix} \approx \begin{bmatrix} \mathbf{W}_l \\ \\ \mathbf{W}_a \end{bmatrix} \mathbf{H} \tag{2.2}$$

When the total set of vocal commands contains L keywords, $\mathbf{W}$ should count at least L columns, but in practice, some extra D columns are added to $\mathbf{W}$ to model the filler words.

The representation of the keywords in $(\mathbf{W}_a)$ can be found by minimizing the Kullback-Leibler divergence between both sides of Equation 2.2,

$$(\mathbf{H}^*, \mathbf{W}_a^*, \mathbf{W}_l^*) = \arg \min_{(\mathbf{H}, \mathbf{W}_a, \mathbf{W}_l)} D_{KL}(\begin{bmatrix} \mathbf{V}_l \\ \\ \mathbf{V}_a \end{bmatrix} || \begin{bmatrix} \mathbf{W}_l \\ \\ \mathbf{W}_a \end{bmatrix} \mathbf{H}) \tag{2.3}$$

When the acoustic representations of keywords $(\mathbf{W}_a^*)$ are learned, keyword recognition can be tested on a test set consisting of unseen (unlabelled) utterances. We denote the data matrix $\mathbf{V}$ of the unseen utterances in the test phase by $\mathbf{V}_t$ and it only contains the acoustic representations of the spoken utterances. The matrix $\mathbf{H}$ in the test phase is denoted by $\mathbf{H}_t$. To recognize the keywords in $\mathbf{V}_t$, $\mathbf{H}_t$ is optimized in order to minimize the distance between $\mathbf{V}_t$ and $(\mathbf{W}_a^* \mathbf{H}_t)$.

$$\mathbf{H}_t^* = \arg \min_{\mathbf{H}_t} D_{KL}(\mathbf{V}_t || \mathbf{W}_a^* \mathbf{H}_t) \tag{2.4}$$

The obtained matrix $\mathbf{H}_t^*$ is used to provide the keyword activation matrix $\mathbf{A}$,

$$\mathbf{A} = \mathbf{W}_l^* \mathbf{H}_t^* \tag{2.5}$$

$\mathbf{A}$ is a $(L \times N)$ matrix and each column in $\mathbf{A}$ corresponds to the respective column in $\mathbf{V}_t$.The higher the score in the rows of $\mathbf{A}$, the more likely that the respective keyword has appeared in the spoken test utterances.

2.5 Label noise

Although supervised NMF approaches ([1, 3, 8–10]) have been studied frequently within the general scope of machine learning (see Section 2.3), to the best of our knowledge, no attention has ever been directed towards the robustness of the approach against label noise. When data is manually annotated, labels are expected to be correct. However, there are numerous ways to end up with labelling errors when a self-learning user interface is trained by the end-user. We consider here four types of label noise: insertion, deletion, substitution and command substitution.

Insertion The command of the user can be underspecified: the command lacks part of the information needed to uniquely determine the intended action of the user. For instance, when the user says: "Turn on" and then turns on the television, the executed action is associated with two keyword labels, "turn on" and "television", while the acoustic input only contains one spoken keyword. There is one additional keyword in the label input compared to the acoustic input.

Omitting a spoken keyword in a command is equivalent to adding a wrong keyword label to a correctly labelled utterance. Because it is difficult to adapt a spoken utterance in an existing speech corpus, we simulate this error by activating a keyword label in $\mathbf{V}_l$ that was not present in the transcription of the spoken utterance. In different words, we increase the frequency count by one in $\mathbf{V}_l$ for a keyword that we simulate to be missing in the spoken command. We call this error an insertion.

Deletion A second kind of error is the over-specification of a spoken command. For instance, the user says "Turn on, the radio, uh no, the television". The acoustic input contains one keyword more than the label input, i.e. "radio". Decreasing a non-zero entry in $\mathbf{V}_l$ by one allows us to simulate the occurrence of one extra keyword in the spoken command. We call this error a deletion.

Substitution The user might mistakenly say "Turn on the radio" but then turns on the television. As a consequence, the label part and the acoustic part of $\mathbf{V}$ are sharing one keyword, i.e. "Turn on", but they are not sharing the second keyword. This error is simulated by one deletion

followed by one insertion in the same column of the label matrix $\mathbf{V}_l$, i.e. in a correctly transcripted utterance. We call this error a substitution.

Command substitution Finally, the user might push the wrong button on a manual user interface. For instance, the end-user might ask his/her partner to switch off the lights because he/she would like to watch television. In the meantime, the user might turn on the television. A total mismatch between the voice command "Switch off the lights" (the acoustic input) and the executed command "Turn on, television" (the keyword labels) is then expected. We simulate this error by taking a correctly annotated utterance, and then, changing the complete column in $\mathbf{V}_l$ to a label vector consisting of a few randomly selected keywords. We call this error a command substitution.

2.6 Experimental setup

2.6.1 Speech data

The speech data was selected from the English corpus constructed in the second year of the ACORNS project [13]. The database consists of $13,160$ utterances, produced by 10 speakers. Each utterance consists of 1 to 4 different keywords embedded in a carrier sentence with unrelated filler words. In total, there are 50 unique predefined keywords. An example of a sentence is presented in Figure 2.1 and the keywords are underlined. Note that, although the speech does not resemble a command and control task, this makes no difference for the purpose of evaluating NMF learning since the size and complexity of the data is similar to, for example, a home automation task. Similar to [1], $9,821$ utterances were randomly selected to compose the training set and $3,268$ utterances were selected to compose the test set. The training set and the test set contained utterances of all 10 different speakers.

2.6.2 Feature extraction

The feature extraction (see Figure 2.1) was done with the Hamming window of 25 ms size and frame shift of 10 ms. Mel-band spectral magnitudes were converted into a 39 dimensional feature vector: 12 Mel Frequency Cepstral Coefficients (MFCC's) plus the frame's log energy and the respective velocity (Δ) and acceleration ($\Delta\Delta$) vectors. The MFCC features were mean and variance normalized. In an intermediate step, the frame-based features of each utterance were transformed into a single histogram represented by one column vector

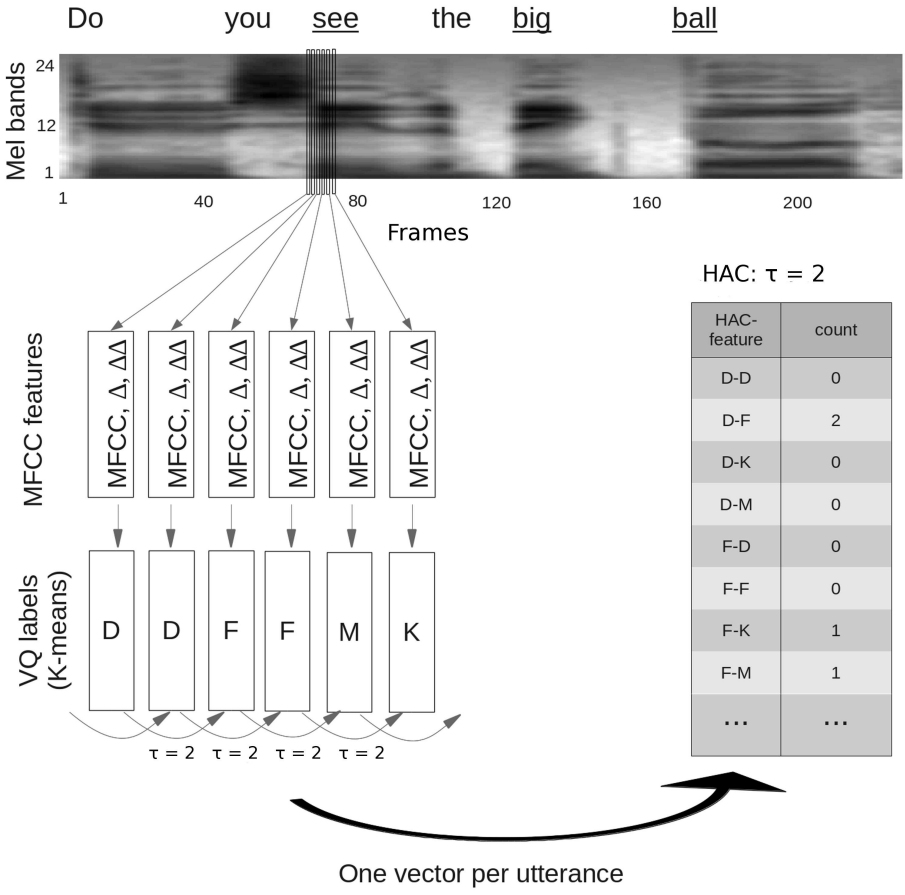


Figure 2.1: *The preprocessing and the feature extraction method*

in $\mathbf{V}_a$. To vectorize the acoustics of the utterances into successive columns, K-means clustering ([14]) was performed on the frames and the cluster centers were used as a Vector Quantization (VQ) codebook of size K. The frame-based features of the whole utterances were then converted into a sequence of VQ labels. The co-occurence of all pairs of VQ labels were counted and these counts were ordered to form the Histogram of Acoustic Co-occurrence (HAC, [1, 10]). More precisely, HAC representations are built by counting the co-occurrence of two VQ labels in different frames over a particular time offset τ between frames. In the toy example of Figure 2.1, each feature vector is clustered and the letters

'D','F','M' and 'K' represent the VQ labels of the clusters. The co-occurence of the VQ-labels in each utterance is counted over a time delay $\tau = 2$. In the experiments, the utterance-based feature vectors consisted of co-occurrence counts of VQ labels for three different codebook sizes: $K = 20, 100$ and 400. Additionally, three different time offsets were used: $\tau = 20, 50$ and 90 ms. Co-occurrence counts for codebook sizes $K = 20, 100$ and 400 for three time offsets resulted in $3 \times (20^2 + 100^2 + 400^2) = 511,200$ features in each column vector of $\mathbf{V}_a$.

2.6.3 Experiment

We tested label noise robustness for different sizes of the training sets: $N = 100$, 200, 500, 1,000, 2,000, 4,000 and 9,821 utterances. Each utterance is considered one training example. For each type of label noise, we had label noise affecting 0, 10, 30, 50, 70 and 90 percents of the utterances in the training sets.

We took some precautions to limit the variation of the experimental results due to the random selection of utterances. First, the smaller training sets were nested in the larger training sets. For instance, the first 100 utterances were selected randomly to compose the training set of $N = 100$ utterances, then, 100 more utterances were selected randomly and added to the first 100 utterances to compose the training set of $N = 200$ utterances. Second, a similar procedure was followed for adding label noise. We first selected 5% of the utterances randomly, then, an additional 5% was selected randomly and added to the first selection to create the condition of 10% label noise. Additionally, to prevent that the results would depend on one particular random selection of utterances, the whole procedure was repeated five times with different random selections of utterances.

The experimental results depend on the initialization of $\mathbf{W}$ and $\mathbf{H}$. We used the same initialization procedure as in [14]. In short, $\mathbf{H}$ was initialized by adding random variation to $\mathbf{V}_l$ and $\mathbf{W}_l$ was initialized by adding random variation to the identity matrix for the first $L \times L$ entries ($L = 50$). In addition, 25 columns were introduced in $\mathbf{W}$ to represent labelless filler words. All other entries in $\mathbf{W}$ were randomly initialized. To control for the influence of initialization, NMF training was repeated five times, each time with different random initializations. In short, we investigated 4 types of label errors with 7 sizes of training sets and 6 levels of label noise. We repeated all experiments ($5 \times 5 =$) 25 times with different NMF initializations and random selections of the training sets.

We measured the effect of NMF training with label errors on keyword recognition in the test set. The test set always consisted of the same utterances. Contrary to common word recognition tasks, word recognition in the current experiments

only involved the detection of a few keywords, ignoring the filler words. When a test utterance contained r keywords, we compared the predictions based on the r highest scores in $\mathbf{A}$ (see Section 2.4) with the correct r keywords in the test utterance. The proportion of correctly recognized keywords against the total number of spoken keywords was defined as the accuracy.

2.7 Results

The resulting accuracies are shown in Figure 2.2. For each type of label error, there is one graph showing the mean recognition accuracy as a function of the

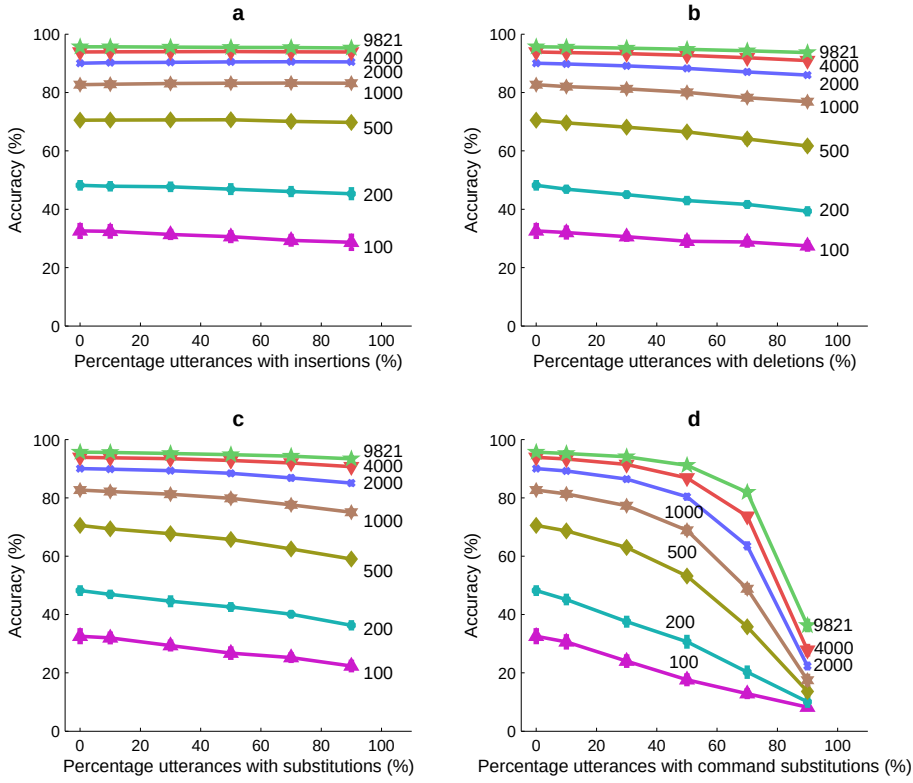


Figure 2.2: *Mean recognition accuracy as a function of training set size and percentage of affected utterances in the training corpus for each label error type:* ***a*** *insertions,* ***b*** *deletions,* ***c*** *substitutions,* ***d*** *command substitutions*

percentage of utterances affected with label noise in the training corpus. For each training set size, there is one trend line and the training set size is indicated at the end or on top of the trend line. The error bars denote the standard errors. In a system that is robust against label noise, the performance is not degrading too much as a function of label noise. Accordingly, we can observe that the proposed method is very robust against label insertions, deletions and substitutions since the lines are nearly horizontal over the whole range. For the largest training set size and without label noise, the performance was 95.6% correctly identified keywords. In the case of 90% utterances affected by label noise, the accuracies were only slightly lower, 95.2%, 93.7% and 93.4% respectively, for the insertions, deletions and substitutions.

However, the curves are more rapidly descending for command substitution errors. Given the definition of the four error types, this difference can be expected because command substitution are affecting all keyword labels in an utterance and not just one. For instance, 90% of the utterances with one insertion, deletion or substitution error corresponds with 31.3% of keyword labels that were affected in the training sets. Contrary to insertion, deletion and substitution, 90% of the utterances with command substitution errors corresponds to 90% of keyword labels that were affected in the training sets. The difference with the former types of label noise demonstrates that the effect of label noise depends on the number of wrong keyword labels rather than on the number of affected utterances.

The second issue of interest is the speed of learning. The higher the accuracies for small training sets, the faster the system picks up the keywords. After 100 correctly labelled utterances of training, which corresponds to an average of 5.6 examples per keyword, the self-learning vocal user interface picked up the associations with an average accuracy rate of 34%. However, 2,000 utterances, which corresponds to an average of 112 examples per keyword in the training set, were needed to reach a performance close to 90%.

2.8 Discussion and conclusion

In the current study, we investigated the robustness of supervised NMF training against different types of label noise. The second aim of the experiments was to test the speed of learning. Our experiments showed that supervised NMF training [1, 10] is very robust against label errors. In practice, that means that an end-user can make many mistakes while training a self-learning NMF-based interface before the performance starts to degrade. Although less label noise robustness has been demonstrated for command substitutions, it is still acceptable for application purposes given that we do not expect more than 30%

to 40% command substitution errors to occur in the interaction between the non-expert user and the vocal user interface.

No explicit method was introduced to obtain these levels of label noise robustness, rather, label noise robustness seems to be an inherent property of supervised NMF. To tentatively explain label noise robustness in supervised NMF, we need to consider the variable nature of speech signals and the way our supervised NMF approach deals with this variation. The spoken utterances and the keywords are represented by a distribution of acoustic features. In NMF learning, acoustic feature distributions for words and their best linear combinations are sought in order to compose all the extracted distributions from the spoken utterances. When a part of the keywords are mislabelled during training, the acoustic feature distributions of these keywords will be composed of an approximately linearly weighted combination of the good and (fewer) bad acoustic examples. During decoding, the affected distributions of the keywords might lower the activation scores, but the highest activations are possibly still corresponding to the spoken keywords in the utterances.

The end-user can permit to make many label errors before the performance starts to degrade, but he still has to put some effort in training the device. When end-users have to demonstrate the meaning of $2,000$ commands to reach a performance of 90%, it looks like a very demanding task. It is difficult to determine an acceptable lower bound for the learning speed of a vocal user interface. Nevertheless, the faster the better, as more users will be prepared to keep on training the device until some comfortable level of service is experienced by the user. Therefore, there is still room for improvement concerning the speed of learning.

There are some suggestions that are helpful to improve overall accuracy rates and speed up the learning curve. First, and most importantly, in the applied set-up, the keywords are learned individually without sharing acoustic sub-structures or patterns besides code books. Unlike humans and state-of-the-art automatic speech recognition systems, there is no phone-level acoustic model that is shared across words. Such a model is required to be able to learn compact lexical-type word descriptions, which would be possible from small training sets. Secondly, it should be noted that the speech data was produced by 10 different speakers, males and females. Since the targeted vocal user interface, however, is self-learning, it aims at learning the speech of only a single user. Keyword recognition accuracies are higher when the experiments are conducted on the data of the individual speakers.

To conclude, the occurrence of label noise is not an issue in a self-learning command-and-control application built on supervised NMF. The first requirement for creating a user-friendly learning environment is therefore met.

However, it would be desirable to have a higher learning speed. More attention to speed up the learning curve is therefore required in subsequent chapters.

2.9 References

[1] J. Driesen, J. Gemmeke, and H. Van hamme, "Weakly supervised keyword learning using sparse representations of speech," in *Proc. ICASSP*, (Kyoto, Japan), pp. 5145–5148, 2012. pages 44, 45, 46, 47, 48, 49, 50, 51, 54

[2] L. ten Bosch, H. Van hamme, and L. Boves, "A computational model of language acquisition: focus on word discovery," in *In Interspeech 2008*, Citeseer, 2008. pages 44

[3] J. Driesen and H. Van hamme, "Modelling vocabulary acquisition, adaptation, and generalization in infants using adaptive bayesian plsa," *Neurocomputing*, vol. 74, pp. 1874–1882, 2011. pages 45, 46, 49

[4] T. Heinroth, M. Grotz, F. Nothdurft, and W. Minker, "Adaptive speech understanding for intuitive model-based spoken dialogues," in *Proc. LREC*, pp. 1281–1288, 2012. pages 45

[5] R. Taguchi, N. Iwahashi, K. Funakoshi, M. Nakano, T. Nose, and T. Nitta, "Learning physically grounded lexicons from spoken utterances," in *Human Machine Interaction - Getting Closer* (M. Inaki, ed.), pp. 69–84, 2012. pages 45

[6] J. van de Loo, J. F. Gemmeke, G. De Pauw, J. Driesen, H. Van hamme, and W. Daelemans, "Towards a self-learning assistive vocal interface: Vocabulary and grammar learning," in *Proc. of the workshop Speech and Multimodal Interaction in Assistive Environments (SMIAE)*, 2012. pages 45

[7] J. Bootkrajang, "Learning with labeling errors," Tech. Rep. CSR-11-07, School of Computer Science, University of Birmingham, April 2011. pages 46

[8] J. Driesen, L. ten Bosch, and H. Van hamme, "Adaptive non-negative matrix factorization in a computational model of language acquisition," in *Proc. Interspeech*, (Brighton, UK), pp. 1711–1714, 2009. pages 46, 49

[9] H. Lee, J. Yoo, and S. choi, "Semi-supervised nonnegative matrix factorization," *IEEE Signal Processing Letters*, vol. 17, pp. 4–7, 2009. pages

[10] H. Van hamme, "Hac-models: a novel approach to continuous speech recognition," in *Proc. Interspeech*, (Brisbane, Australia), pp. 255–258, 2008. pages 46, 47, 48, 49, 51, 54

[11] D. Lee and H. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999. pages 47

[12] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proc. ACM SIGIR Conf. Research and Development in Information RetrievalInterspeech*, (Toronto, Canada), 2003. pages 47

[13] L. Boves, L. ten Bosch, and R. Moore, "Acorns-towards computational modeling of communication and recognition skills," in *Proc. IEEE int. Conf. On Cognitive informatics*, (California, USA), pp. 349–355, 2007. pages 45, 50

[14] J. Driesen, *Discovering words in speech using matrix factorization.* PhD thesis, K.U.Leuven, ESAT, July 2012. pages 51, 52

Chapter 3

Speaker-dependent and acoustic procedures for NMF

This chapter is based on the following article:
B., Ons, J.F., Gemmeke, H., Van hamme, "Fast vocabulary acquisition in an NMF-based self-learning vocal user interface," **Computer, Speech & Language,** vol. 28, no. 4, pp. 997 - 1017, 2014.

3.1 Abstract

In command-and-control applications, a Vocal User Interface (VUI) is useful for handsfree control of various devices, especially for people with a physical disability. The spoken utterances are usually restricted to a predefined list of phrases or to a restricted grammar, and the acoustic models work well for normal speech. While some state-of-the-art methods allow for user adaptation of the predefined acoustic models and lexicons, we pursue a fully adaptive VUI by learning both vocabulary and acoustics directly from interaction examples. A learning curve usually has a steep rise in the beginning and an asymptotic ceiling at the end. To limit tutoring time and to guarantee good performance in the long run, the word learning rate of the VUI should be fast and the learning curve should level off at a high accuracy. In order to deal with these performance indicators, we propose a multi-level VUI architecture and we investigate the effectiveness of alternative processing schemes. In the low-level layer, we explore the use of MIDA features (Mutual Information Discrimination Analysis) against conventional MFCC features. In the mid-level layer, we enhance the acoustic representation by means of phone posteriorgrams and clustering procedures. In the high-level layer, we use the NMF (Non-negative Matrix Factorization) procedure which has been demonstrated to be an effective approach for word learning. We evaluate and discuss the performance and the feasibility of our approach in a realistic experimental setting of the VUI-user learning context.

3.2 Context and contributions of the chapter

Fast learning during usage is crucial in our self-taught VUI approach. The VUI models develop and improve as long as users keep demonstrating the meaning of their spoken commands. A common cause why users stop using their system is because of unsatisfactory return. This is a vicious cycle: unsatisfying returns leads to a demotivated user who provides less and less demonstrations, which leads to limited training data and low accuracy. The low accuracy completes the vicious cycle and leads to unsatisfying returns from the VUI.

Therefore, it is important that the system learns from a few examples. In this chapter, we describe a few principles that are useful to improve the learning speed. The contributions are the following. First, we demonstrate fast learning for normal speech by using phone posteriorgrams. Second, we demonstrate that the results are largely depending on the development set from which phone models or vectorquantized clusters are created. Third, we demonstrate better performance for a system with speaker-dependent acoustic features and with speaker-dependent NMF data. Thus, a considerable gain was obtained

by acquiring speaker-dependent acoustic models and speaker-dependent NMF models. This finding has a significant influence in subsequent chapters since we developed speaker-dependent procedures from that moment on.

The ALADIN home-made corpus was in progress but not yet available at the time of writing the original publication. Therefore, we used the corpus with normal speech from the ACORNS project. We were aware of the fact that normal speech is not a good match for dysarthric speech. Nevertheless, the VUI should be able to handle normal speech as well. By building speaker-dependent models, we obtained a baseline with a considerable improvement in learning speed and accuracy for evaluating further research.

3.3 Introduction

Command-and-control (C&C) speech recognition allows users to interact with systems like domestic devices, assistive technology, computers, smart-phones or other mobile devices. The user speaks a command or a phrase to control different functions in the environment like the central heating or the light units in the house, to retrieve information on their smartphone or to navigate through a menu on a computer. C&C applications are especially useful for people with a physical disability affording them handsfree control of their wheel chair, the positioning of their bed or other independent living aids.

In most speech driven C&C applications, the spoken commands are restricted to a predefined list of phrases described by a restricted grammars and vocabulary. The size of the vocabulary ranges from a few to a few hundred words and the grammars are mainly rule-based. Although the targeted VUI application allows a developer to consider many interaction scenarios beforehand, the use of a VUI is not always successful when the interaction oversteps the clear boundaries of the lexicon, the grammars or the dialogue models. Even in less restrictive frameworks, such as in the now popular Siri speech recognition application for the iPhone, performance degrades rapidly if the acoustic models do not match the speech material used to train the system, for example on accented or dysarthric speech. The goal of this paper is to investigate a VUI model which is able to associate any utterance to a C&C action allowing command and control usability by deviant speech as well.

Over the past decade, various approaches have been proposed for adaptation to unexpected circumstances in real-life situations. For instance, [1] proposed a statistical model for mobile devices that tracks the past of the user's behaviour in order to predict commands. In [2], grammars were able to adapt dynamically to real-life communication making interactions more natural. In [3] and [4],

speaker-independent acoustic models were adapted to speaker-dependent models allowing for better recognition of the user-specific vocalizations. There are plenty more studies that paved the way to more natural interaction with machines and devices by means of human-centred design and user adaptation. For instance, in a study of [5] a robust speech recogniser was developed to adapt to dysarthric speech as well. In the "Speech Training And Recognition for Dysarthric Users of Assistive Technology" (STARDUST) project [5], the problem was tackled by adaptation in two directions: a training package assisting dysarthric speakers to improve the recognition likelihood of their utterances (users adapting to speech recognition systems) and speech recognition systems having greater tolerance to variability of dysarthric vocalizations (speech recognition models adapting to users) were developed.

However, all these approaches have in common that these systems are still based on acoustic and language models that are trained beforehand and adapted through interaction to the spoken utterances of the user. While these methods focus on adaptation, we focus on *grounding*: learning both vocabulary and acoustics directly from the user during the usage of the VUI. The grounding process [6] refers to the process by which common ground or meaning is built between the user and the system. Situated in the "Adaptation and Learning for Assistive Domestic Vocal Interfaces" (ALADIN) project [7, 8], we aim to design a VUI that learns to understand speech by mining the speech input from the end user and the changes that are provoked on a device.

The VUI should learn to understand classes referring to devices, actions or properties by using cross-situational evidence and learning the statistical regularities between two modalities, namely, the spoken utterances of the user and the feedback coming from the device(s). Supervision coming from the device is weak in the sense that the information provided to the VUI consists of signals referring to states and actions in a machine without any chronological information, orthographic nor phonetic transcriptions. Earlier studies have demonstrated that multi-modal Non-negative Matrix Factorisation (NMF) is a useful tool to learn weakly co-occurring regularities over two modalities in order to find the intra- and inter-modality patterns. For instance, in [9], NMF is used to generate multimodal image representations that integrate visual and text features for image collections guided by ratings, comments and tags on the web. [10] used a similar approach and called it multiview clustering to cluster images and predict image labels. Similar to NMF-based keyword discovery in [11], we use NMF to learn co-occurrences between acoustic feature vectors emerging from the spoken utterances and semantic label vectors describing the action properties.

In order for a self-learning approach to be useful as a VUI, the learning process should be *fast*. At the same time, after sufficient training tokens have been

presented, the *accuracy* should be high. The contribution of this work is twofold. First, we investigate to what extent the learning speed and accuracy can be improved by using more advanced feature representations in NMF. We use phone classifiers to create phone confidence measures to replace the conventional acoustic input in NMF learning [12, 13]. In addition to phone classifiers, we also evaluate a speaker-dependent version of soft Vector Quantization (soft VQ), which is a data-driven and probabilistic procedure to cluster the acoustic data of the speaker. We tested the usefulness of this data-driven approach for small training sets as user-specific data is expected to be scarce in the beginning of the VUI usage.

A second contribution of this work is that we investigate to what extent the NMF machine learning procedure can be used for a VUI under realistic constraints. While previously, NMF evaluations were typically speaker-independent, we will work speaker-dependent since the system is self-learning and builds its representations from scratch. Also, while there is some prior work on investigating the learning speed of NMF [14–16], this made unrealistic assumptions on the amount of speech material available during training for building the lower-level acoustic representations. Here, we will use speaker-independent material from different annotated datasets to train the phone classifiers or VQ clusters beforehand and use only speaker-dependent training data in proportion to the expected accumulative production of speech in a real VUI-user learning context, therefore, evaluating the feasibility of the VUI by limiting access to available data corresponding to a realistic operating mode.

The remainder of the chapter is organised as follows. In sections 3.4 and 3.5 we introduce the learning framework, including the feature representations, acoustic models and NMF procedure used throughout the paper. In section 3.6 we conduct a series of experiments to evaluate the effectiveness of the NMF approach on the ACORNS database [17] containing normal speech. NMF learning has been evaluated on ACORNS data [e.g. 12] and therefore we use ACORNS as well to introduce a proper baseline. We discuss related work and present our thoughts on future work in section 3.7.

3.4 A self-taught user interface

3.4.1 The learning problem

Self-learning refers to the VUI's ability to learn from interactions with the end user. The training token consists of speech paired with the demonstration of the intended action. For instance, the user utters the command: "Close the door, please" and the VUI forwards that command to the home automation system.

However, if the VUI is lacking confidence, the user is asked to demonstrate the intended action, for instance, by pushing the door-closing button on the control panel. The VUI infers the executed action from information sent by the control device. This assumes that a number of properties and actions enabling the control of a device are predetermined and that a placeholder is provided for each one of them. During training, the acoustic representation of the spoken words is associated with this control information. The user's command and the demonstrated action is counting as one training example. If the VUI parses the wrong command, then the user has the opportunity to overrule the action. The overruling action will then serve as grounding information. In this study, we evaluate the performance in function of correct demonstrated actions and spoken commands.

In Table 3.4.1, the learning problem is demonstrated by means of a toy example. Supervision in Table 3.1a is displayed by the pictograms representing semantic tags like a device, an activity, or a property produced by a button-push. Assuming that the acoustic representations of the spoken utterances are represented by the text characters and the semantic tags by the pictograms in respectively the first and the second column of Table 3.1a, then the learning process consists of finding the recurrent acoustic patterns (at least two adjacent letters) and their co-occurring semantic tags that make up the discriminative parts of the user's commands. These recurring patterns are displayed in Table 3.1b.

3.4.2 Non-negative matrix factorization

Learning a word by means of the statistical co-occurrence of multimodal evidence across situations is called cross-situational learning [18]. In earlier studies [11, 14, 19], weakly supervised Non-negative Matrix Factorization (NMF) has been presented as a useful machine learning procedure to discover and learn the acoustic representation of words accompanied by weak supervision. NMF works by factorizing a collection of utterance-based representations into the product of a matrix containing the latent factors describing the recurrent acoustic patterns (such as words) in utterances, and a matrix describing for each utterance which latent factors are active. In weakly supervised NMF, the utterance-based representations are accompanied by grounding information, i.e. label vectors referring to the semantic tags, and the aim is to find the recurrent patterns that co-occur with the semantic tags in the first matrix of the factorization and the activations of these semantic tags in the second matrix component (see Table 3.4.1).

Phrases	Tags
look, *I'm* ***clos****ing the* ***window***	▶◀ ▦
door clos*e*, please	▶◀ ▯
I'm ***open****ing the* ***door*** *now*	◀▶ ▯
open *the* ***window***	◀▶ ▦
switch on *the* ***TV***	○ 📺
the ***heating*** system should be ***turn****ed* ***off***	— ▥
I ***turn off*** *the* ***TV***	— 📺
I ***switch on*** *the* ***heating****, now*	○ ▥

(a) VUI training samples

▶◀	▦	◀▶	▯	—	📺	○	▥
c	w	o	d	t	T	s	h
l	i	p	o	u	V	w	e
o	n	e	o	r		i	a
s	d	n	r	n		t	t
	o			o		c	i
	w			f		h	n
				f		o	g
						n	

(b) Associations with semantic tags

Table 3.1: *The learning problem. The letters in the top table represent the acoustic signal, italic text indicates a recurrent pattern and the bold text represents the co-occurrence with the semantic tags that are displayed in the second column. From the cross-situational evidence, the acoustic feature representation for each semantic tag displayed in the bottom table should be learned*

3.5 Architecture

3.5.1 Overview

In this study, we investigate whether different preparations of the data lead to different learning rates in NMF. We consider two types of training sets, the first type is called the *keyword-learning training set* and it contains the so-called correctly supervised learning examples occurring in our simulated VUI-user usage context. The keyword-learning training set is used to build the NMF model, i.e. latent acoustic representations for C&C keywords, and in this study it is always based on the ACORNS database.

Some steps in the processing flow use acoustic models to prepare the feature vectors for NMF and these NMF-supportive models require training too. For instance, a phone recogniser usually needs a phone-based hidden Markov model (HMM) and training involves annotated speech data. These supportive models

are trained using the second type of training set which is called *acoustic-model training set*.

A schematic overview of the learning framework studied in this chapter can be found in Figure 3.1. Here, the processing takes places from bottom to top, and the multiple directed arrows indicate various combinations of processing steps. First, the spectro-temporal features are extracted from the speech signal (section 3.5.2), resulting in either Mel Frequency Cepstral Coefficients (MFCC, c.f. section 3.5.2) or Mutual Information Discriminant Analysis features (MIDA, c.f. section 3.5.2). The horizontal arrow leading to the MIDA features from the left indicates that for the creation of MIDA features, annotated speech material is needed. In the next step, the spectro-temporal features are converted into posteriorgrams (section 3.5.3), either by using soft-VQ clustering (c.f. section 3.5.3) or a phone recogniser (c.f. section 3.5.3). As for the MIDA feature extraction, the horizontal arrow leading to the phone recogniser from the left indicates that for this approach, annotated speech material is needed. The horizontal arrow leading to the soft VQ procedure from the right indicates that speech material is needed to train the code books. Finally, the posteriorgrams are converted to utterance-level representations (section 3.5.3) by using Histograms of Acoustic Co-occurrence (HAC) after which the NMF training takes place (section 3.5.4). The horizontal arrow leading to the NMF training phase visualises the fact this training is supervised with grounding information.

3.5.2 Feature extraction

MEL spectrum

Speech samples are transformed into Mel-spectra. We used a short time Fourier transform with the Hamming window of 25 ms and a frame shift of 10 ms, followed by a bank of Mel-scaled triangular filters. There are two alternatives in feature extraction resulting in either MFCC or MIDA features.

MFCC features

MFCC features [20] are obtained by applying the Inverse Discrete Cosine Transform (IDCT) to the log-Mel spectral representation. The IDCT expresses the signal in terms of a sum of orthogonal cosine functions oscillating at different frequencies and their amplitudes correspond to the cepstral coefficients in MFCC features. The cepstra of the first 12 cosine functions ($c_0 \dots c_{11}$) and the log energy are retained. The 13-dimensional representation is augmented with their

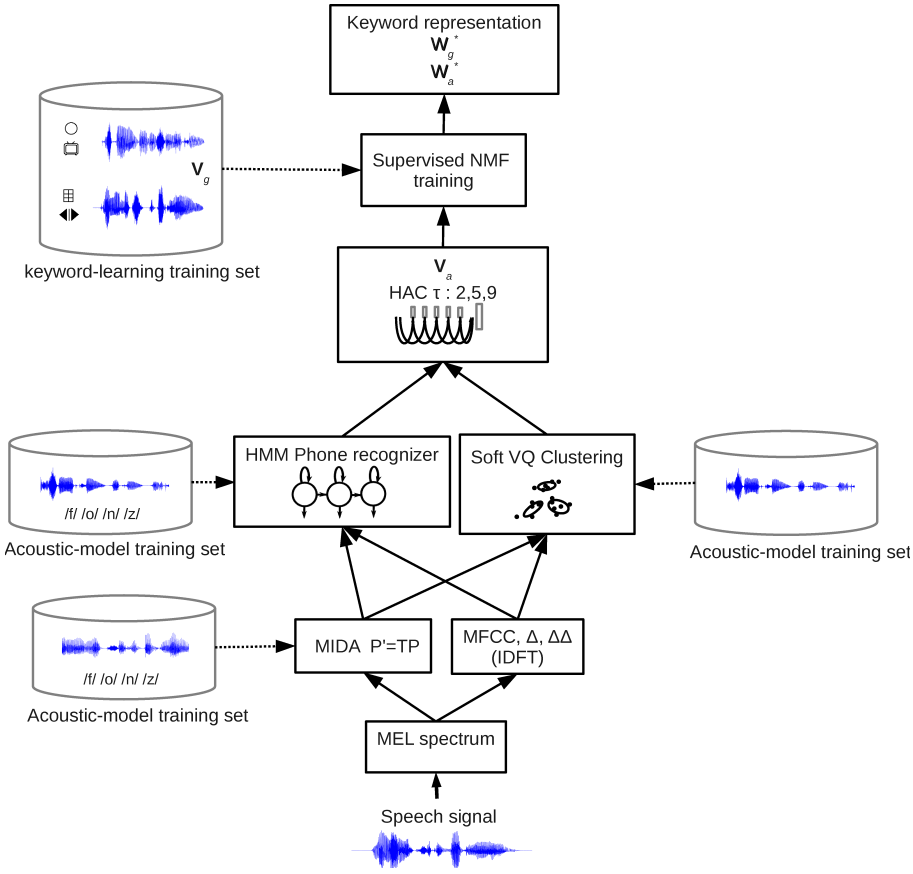


Figure 3.1: *Multi-layered architecture. The architecture allows to compose high-level acoustic units and facilitates associative learning between grounding information and acoustic vectors. The steps depicted at the same level are denoting exchangeable alternatives.*

first and second order differences (Δ- and $\Delta\Delta$-features), yielding a total of 39 coefficients per frame. The MFCC features are mean and variance normalized per utterance.

MIDA features

Discriminant analysis algorithms for feature extraction are often based on a transformation which maximizes the between-class scatter and minimizes the within-class scatter. In Mutual Information Discriminant Analysis or MIDA [21], a linear transformation is sought that maximizes the mutual information between the transformed features and the target classes. The algorithm uses frame-based target class annotations. In our study, the target classes consist of phones, but other annotations can be used as well such as VQ clusters [see 11]. The MIDA transformed features, or MIDA features in brief, are a linear combination of 22 log-MEL spectral dimensions and their first and second order differences (Δ and $\Delta\Delta$). The MIDA features are ordered in terms of mutual information, and we reduced the dimensionality to the 39 most informative MIDA features to conform with the dimensionality of the MFCC features.

3.5.3 Posteriorgram

The feature vectors obtained by the different feature extraction methods are transformed into a posteriorgram. A posteriorgram is a matrix with one dimension pertaining to an exclusive set of acoustic units, denoted by Φ, and the second dimension pertaining to the sequence of frames, denoted by t_i with $i = 1 \ldots Q$ and Q the number of frames. The posteriorgram contains the posterior probabilities, denoted by $\mathbf{P}_{t_i,\theta}$, that the observation in the frame at time t_i originated from an acoustic unit, denoted by θ ($\theta \in \Phi$). The dimensionality of a posteriorgram is $L \times Q$ with L the number of exclusive acoustic units.

We evaluate two alternative acoustic units: Soft VQ clusters [12, 22] modelled by one Gaussian for each cluster in the feature space and phones modelled by a tri-state HMM. Both alternatives are usually based on models with parameters that are estimated beforehand.

Soft VQ posteriorgram

First, clusters are obtained by a codebook training procedure adopted from [12] and [22]. The code book training procedure starts with one cluster joining all frames, and the cluster(s) are split iteratively in sub-clusters until the requested number of clusters is obtained. The iterations comprise two steps. In the first step, the cluster with the largest covariance is split into two clusters by replacing the centre of the respective cluster with two new centres located in the neighbourhood of the old one but shifted in opposite directions along the

main axis of variation. In the second step, k-means clustering is applied using 15 iterations in which frames are partitioned into clusters based on the shortest distance to the centres, and then, centres are estimated for the new clustered frames. In the reported experiments, different code books were used of different sizes: $L = 20, 100$ and 400.

Secondly, all frames in the training set are partitioned in the obtained clusters and a full covariance Gaussian is estimated on all the frames that fall in each respective cluster.

During decoding, a posteriorgram for each frame is obtained by evaluating the probability density functions of the Gaussians at the location of the frame in the feature space and by normalizing the relative likelihoods of the Gaussians to one, i.e. representing each frame by a multinomial probability distribution where each entry denotes the chance that the frame-based observation was emitted by the respective Gaussian. We refer to the cluster-based posteriorgram with the term "soft VQ representation".

Phone posteriorgram

An alternative for soft VQ representations are phone posteriorgrams. First, a phone recogniser is built by training tri-state HMM mono-phone models based on a training set with speech data and phonetic transcriptions for a particular phone alphabet with a size of L phones. A mixture model with G diagonal-covariance tied Gaussians is used to model the observation probabilities of the phone states.

In the decoding phase, the acoustic models and HMM topologies are used to build a directed acyclic graph, building on the ten best Viterbi scores [23] in which each arc represents a phone. For each time frame, we accumulate the scores of the arcs passing the frame for each respective phone and we calculate posterior probabilities by using the forward-backward algorithm [see 24]. By normalising the accumulated scores, we obtain posterior probabilities.

Note that we do not use phone n-gram relations, to avoid the risk that the posterior probabilities are influenced by the average co-occurrence patterns in the data instead of reflecting the instantaneous acoustic sounds as such.

Histogram of acoustic co-occurrence

The posteriorgram of an utterance has a variable length that depends on the number of frames in an utterance. However, fixed-length vectors are required to compose the data matrix for NMF. The aim of HAC [11, 19, 25] is to build

a fixed-length vector for each utterance by accumulating the probability of observing a phone or a VQ cluster pair (α, β) for all possible $L \times L$ pairs over two frames shifted τ frames away from each other. The probability of co-occurrence can be accumulated over the whole utterance for every possible phone or VQ cluster pair resulting in a fixed length vector of $F = L^2$ entries. For utterance n having q sequential frames, the co-occurrence score for the phone or the VQ cluster pair (α, β) with $\alpha, \beta \in \Phi$, and Φ the phone or code book set, can be expressed as follows

$$[\mathbf{v}_n^\tau]_{(\alpha,\beta)} = \sum_{t_i=0}^{q-\tau} \mathbf{P}_{t_i,\alpha} \mathbf{P}_{t_i+\tau,\beta} \tag{3.1}$$

and $\forall t_i, i = 1 \dots Q, \sum_{\theta \in \Phi} \mathbf{P}_{t_i,\theta} = 1$.

An utterance is then represented by an accumulation of phone or soft VQ co-occurrence probabilities. In the current experiment we used different τ lags with $\tau = 2, 5$ or 9 [26, 27]. Each utterance is represented by a single fixed-length column vector $\mathbf{v}_{a,n}$,

$$\mathbf{v}_{a,n} = \begin{bmatrix} \mathbf{v}_n^{\tau=2} \\ \mathbf{v}_n^{\tau=5} \\ \mathbf{v}_n^{\tau=9} \end{bmatrix} \tag{3.2}$$

in which all combinations $(\alpha, \beta) \in \Phi \times \Phi$ are stacked over different time lags for one whole utterance. Additionally, we implemented different code books of different sizes L and the number of code books is denoted by the constant C [16, 28]. All co-occurrence scores corresponding to the C code books are stacked in one utterance-based vector. For the collection of N utterances in the training set, the acoustic representation is denoted by $\mathbf{V}_a = [\mathbf{v}_{a,1} \mathbf{v}_{a,2} \dots \mathbf{v}_{a,n}]$ with $n = 1, \dots, N$.

3.5.4 NMF learning

In supervised NMF learning [11, 19], the acoustic representation $\mathbf{V}_a$ of the training set is augmented with grounding information $\mathbf{V}_g$. In $\mathbf{V}_g$, the presence of keywords in each utterance is indicated as follows: There is one row in $\mathbf{V}_g$ for each keyword label and its entries represent the number of times that the respective keyword was uttered. $\mathbf{V}_a$ is a $(F \times N)$ matrix with F the acoustic feature dimension and $\mathbf{V}_g$ is a $(K \times N)$ matrix with K the number of keywords.

Non-Negative Matrix Factorization will decompose the matrix $\left[\mathbf{V}_g{}^T \mathbf{V}_a{}^T\right]^T$ into the product of two low-rank matrices.

$$\begin{bmatrix} \mathbf{V}_g \\ \\ \mathbf{V}_a \end{bmatrix} \approx \begin{bmatrix} \mathbf{W}_g \\ \\ \mathbf{W}_a \end{bmatrix} \mathbf{H} \tag{3.3}$$

The purpose of the supervised NMF learning is to find the latent acoustic representation for each property or action needed to control a device. The columns in $\mathbf{W}_a$ represent the latent structure, i.e. the recurring acoustic patterns of the columns in $\mathbf{V}_a$ co-occurring with the semantic tags (see section 3.4) that are represented by label vectors in $\mathbf{V}_g$ similar to the keywords in [11, 19]. The columns in $\mathbf{H}$ indicate which patterns, thus columns in $\mathbf{W}$, are combined to approximate the columns in $\mathbf{V}$. When the total set of vocal commands contains K semantic items for which the system needs to learn one word, $\mathbf{W}$ should count K columns, but we add D extra columns in $\mathbf{W}$ to model filler words. Note that filler words can also model synonyms for the first K words on condition that the synonyms are frequently spoken by the user. Another approach to learn synonyms is by detecting the use of a second word for a particular label vector after which a second column in $\mathbf{W}$ is introduced for this label vector. However, the treatment of synonyms is not pursued in this study. We refer to the learned words with the term "keywords".

The representation of the keywords in $(\mathbf{W}_a)$ can be found by minimizing the Kullback-Leibler divergence between both sides of Eq. 3.3,

$$(\mathbf{H}^*, \mathbf{W}_a^*, \mathbf{W}_g^*) = \arg \min_{(\mathbf{H}, \mathbf{W}_a, \mathbf{W}_g)} D_{KL}(\begin{bmatrix} \mathbf{V}_g \\ \\ \mathbf{V}_a \end{bmatrix} || \begin{bmatrix} \mathbf{W}_g \\ \\ \mathbf{W}_a \end{bmatrix} \mathbf{H}) \tag{3.4}$$

Iterative update rules for minimizing a distance measure between the left and the right handside can be found in [11], [29] and [19]. Convergence is guaranteed towards a local optimum.

Keyword recognition can be tested on a test set. We denote the data matrix $\mathbf{V}$ and $\mathbf{H}$ for the factorization of the test set by $\mathbf{V}_r$ and $\mathbf{H}_r$. $\mathbf{H}_r$ is found by minimizing the Kullback-Leibler divergence between $\mathbf{V}_r$ and $(\mathbf{W}_a^* \mathbf{H}_r)$.

$$\mathbf{H}_r^* = \arg \min_{\mathbf{H}_r} D_{KL}(\mathbf{V}_r || \mathbf{W}_a^* \mathbf{H}_r) \tag{3.5}$$

$\mathbf{W}_a^*$ consists of the latent structure found in the training phase, one column vector for each word, and $\mathbf{H}_r$ indicates which words need to combine to approximate the utterance-based data of the representations of the test set $\mathbf{V}_r$. An optimal

solution for $\mathbf{H}_r$ is guaranteed since Eq. 3.5 is a convex problem. The obtained matrix $\mathbf{H}_r^*$ is used to provide the keyword activation matrix $\mathbf{A}$,

$$\mathbf{A} = \mathbf{W}_g^* \mathbf{H}_r^* \tag{3.6}$$

$\mathbf{A}$ is a $(K \times N)$ matrix and each column in $\mathbf{A}$ corresponds to the respective column in $\mathbf{V}_t$. The higher the score in the rows of $\mathbf{A}$, the more activation for the respective keyword in the spoken test utterances.

3.6 Experiments

3.6.1 Overview

The goal of the experiments is to investigate the different processing flows and to evaluate keyword recognition accuracy in the initial and final phase of the learning curve. The learning curve is the curve representing the acquisition of words in function of the number of learning examples for the average user. Usually, the rate of learning is sharpest in the beginning and gradually evens out. In the reported experiments, we show stepwise improvements of the different proposed flows in the architecture demonstrated in section 3.5.

In the first experiment (section 3.6.3), we set a baseline by using soft VQ mid-level representations (see section 3.5.3) and we introduce phone posteriorgrams as a substitute for the existing method of soft VQ representations. Phone posteriorgrams have been used in NMF for the discovery of latent phone patterns [30], but to the best of our knowledge, it has not been used in NMF for the purpose of fast learning. In the second experiment (section 3.6.4), we investigate the difference in performance using MFCC or MIDA features.

Contrary to the first two experiments, where the training material consisted of speech from different speakers, the training material in the third experiment is speaker-dependent. Speaker-dependent keyword training is pursued in section 3.6.5. Improvements by speaker-dependent training are convenient since the VUI is a personalized system. We refer to the third experiment as *user-centred keyword learning*.

In the light of our aim to investigate the feasibility of using NMF in a self-learning VUI, it makes sense to distinguish *realistic* processing flows from *unrealistic* ones. A realistic processing flow is a simulation of the VUI training corresponding to a real-life VUI-user context allowing only the supposed available training data to train the supportive acoustic models. Realistic and unrealistic are closely related to speaker-dependent and speaker-independent models. Speaker-independent

acoustic models using corpora such as those employed in the field of speech recognition are considered realistic since speaker-independent models can be trained beforehand in a lab. However, speaker-dependent supportive models are only considered realistic when the training set contains utterances that have been spoken by the user in our simulated incremental VUI training. If the user has only spoken 10 utterances in our simulation at a particular moment of time, then the training set contains only those 10 utterances. However, since we are interested in the feasibility of using NMF in a personalised VUI system, it is relevant to verify NMF learning using acoustic-model training sets for which training is speaker-dependent and optimal. Such a processing flow is using all data in the corpus, also the supposed unavailable data in the incremental and interactive learning context. It is regarded as an unrealistic processing flow, but it serves as an upper bound for what NMF can achieve when supportive models are optimally trained. In section 3.6.6 and 3.6.7 we progress to more realistic speaker-dependent models. In the fourth experiments (section 3.6.6), we focus on speaker-dependent code-book training. Since the speech of the user is scarce in the beginning of the VUI usage, the speaker-independent models trained on large corpora might outperform the speaker-dependent models with few initial training examples. We refer to speaker-dependent training of code books as *user-centred codebook training*.

When using one processing flow, it is difficult to obtain good results for both performance indicators, i.e. fast learning and high asymptotic accuracy. In the last Experiment 3.6.7, we combine two processing flows that are complementary in performing well on both performance indicators in the preceeding experiments and we evaluate whether the combined flow is able to improve fast learning and high asymptotic accuracy.

3.6.2 Experimental setup

Databases and datasets

Data from different corpora is used to compose the training sets. All corpora contain normal speech. We use the data of the "Acquisition of Communication and Recognition Skills projects", *ACORNS* [17], to represent the commands that the end user utters to train the VUI. More particularly, we use the UK English subset of the corpus developed in the second year of the ACORNS project [31]. The subset of the corpus consists of 13160 utterances produced by 10 different speakers: four speakers produced 2396 utterances and six speakers produced 596 utterances. Only the speech data of the first four speakers is selected since the amount of speaker-specific data is important for simulating the asymptotic behaviour of the VUI. Utterances consist of 1 to 4 different

keywords embedded in a carrier sentence with unrelated filler words. In total, there are 50 unique predefined keywords and 30 filler words. The choice for the corpus fits well for the purpose of evaluating the learning curve of the VUI as the size and complexity of the data is similar to a common home automation task. We refer to the ACORNS subsets as the *keyword-learning training sets*.

We investigate fast learning by using keyword-learning training sets of increasing sizes, and we refer to the *series* of gradual increasing training sets with the term *fold*. In each fold the smaller sets are forming (nested) subsets of the larger training sets, i.e. representing snapshots of the same learning curve. The obtained accuracies for small training sets correspond to the accuracies that can be expected in the beginning of the learning curve when the VUI is put into service and the user starts the training. The accuracy of the largest training set corresponds to the accuracy for the case that the user has trained the VUI during a longer period of time. When NMF learning includes multiple speakers, the data is pooled for the different speakers and the mixed training sets count N utterances of all users with $N = 50, 100, 200, 400, 800, 1600, 3200$ and 7156 after excluding 32 utterances due to bad quality. The corresponding test sets count 2382 utterances after excluding 14 utterances. When NMF learning is user-centred, the folds are composed of training sets with training data from individual speakers and set sizes $N = 50, 100, 200, 400, 800$ and $N = 1790, 1786, 1789$ or 1791 for the largest training set of the fold depending on the respective speaker. The sizes of the corresponding test sets are $593, 594, 596$ and 599 utterances for the four speakers, respectively. The average keyword occurrence for all 50 keywords is $3.0, 5.9, 11.8, 23.6, 47.3$ and 105.7 times over all folds for data set sizes $N = 50, 100, 200, 400, 800$ and $N > 1785$, respectively.

We created three different folds with gradually increasing training set sizes for each processing flow under investigation by selecting utterances randomly without replacement. For each fold, $\mathbf{H}$ and $\mathbf{W}$ are estimated five times (see section 3.5.4) using a different initialization (see section 3.6.2) leading to different solution for the same fold. Initializing $\mathbf{H}$ and $\mathbf{W}$ five times for three folds results in 15 learning curves for each processing flow.

We use three different corpora to train the acoustic models at different layers: 1) ACORNS, 2) the "Wall Street Journal corpus recorded at the University of Cambridge, phase 0", *WSJCAM0* [32], which is the UK English equivalent of a subset of the US English Wall street Journal corpus (WSJ0) and 3) the "Corpus Gesproken Nederlands", *CGN* [33], which is a Dutch corpus consisting of continuous speech covering news bulletins selected from Dutch television and radio. ACORNS is a UK English corpus used here to simulate the spoken phrases of the VUI user, so the native language of the user in the training simulation of the VUI is implicitly set to UK English.

Feature dimensionality

The phone posteriorgrams (see section 3.5.3) have dimension $L = 41, 46$ or 50 depending on the size of phonetic alphabets used in the transcriptions of ACORNS, WSJCAM0 or CGN, respectively. The phonetic alphabets also include one noise unit and one silence unit in addition to the phones. The noise and silence units model the silence and the non-speech sounds such as coughts or breathing sounds. The phone models are trained using the open source software SPRAAK [34]. For WSJCAM0, a mixture model with $G = 16822$ diagonal-covariance tied Gaussians is used to model the observation probabilities of the phone states. Similarly, $G = 5813$ and $G = 48845$ tied Gaussians are used for ACORNS and CGN respectively. Phone posteriorgrams are converted to HAC's (see section 3.5.3). The dimension of the HAC feature vector F depends on L and on the number T with T the number of frame-lags τ, $F = T \times (L^2)$ or $F = 5043, 6348$ and 7500 features depending on the training set of the phone recogniser.

The number of code books C is 3 and the dimension L for the code books was freely chosen with $L = 20, 100$ or 400 (see section 3.5.3). Co-occurrence scores for soft VQ for three code books jointly and three frame lags produce fixed-length vectors in $\mathbf{V}_a$ with size $F = 3 \times (20^2 + 100^2 + 400^2) = 511200$ features.

The main portion of the probability mass per frame seems to originate from only a few phones or VQ clusters. We found out that only non-significant gains are obtained by taking more than the three largest probabilities into account for the soft VQ representations and more than the 10 largest probabilities for the phone posteriorgrams. Therefore, we only take the three highest probabilities per frame into account for soft VQ and the ten highest probabilities for phone posteriorgrams leading to $3 \times T \times C = 27$ and $10 \times T = 30$ non-zero entries per column in the posteriorgram for soft VQ and phones, respectively. Phone posteriorgrams usually lead to less sparse NMF problems but their dimensionality is considerably lower.

Naming

The successive steps of the processing flows from the bottom to the top of Figure 3.1 are reflected in the names of the different graphs in each experiment. There are two possible acoustic-model training sets, one for learning the MIDA transformation and one for training the VQ Gaussians or the phone HMM. The processing flows using MFCC features start with "MFCC" and the processing flows using MIDA features start with "MIDA" followed by the training corpus for the MIDA transformation between parentheses: "ACN" for ACORNS,

"WSJ" for WSJCAM0 or "CGN" for CGN. The second part of the names refers to the mid-layer representation "SVQ" for soft VQ and "PHN" for phone posteriorgrams, followed by the name of the training corpus used to train the code books or the phone HMM. An overview is presented in Table A.0 in the Appendix. The NMF procedure and the keyword-learning training sets are identical for the first two experiments. For the remaining three experiments, the keyword training sets are limited to the speech of one speaker. Since keyword learning is identical within each experiment, it is not incorporated in the names of the graphs. In the last two experiments, the data of each individual speaker in ACORNS is also used to train codebooks, then, we add SD for speaker-dependent and SDD for speaker and set-size dependent inside the parentheses. SD means that the training material only consists of utterances from the respective speaker and SDD means that the training material is SD and limited to the keyword-learning training set, that is the set of utterances that has been exposed in the simulated VUI training.

NMF initialization

$\mathbf{H}_{init}$ and $\mathbf{W}_{init}$ denote the initialisation of $\mathbf{H}$ and $\mathbf{W}$ respectively

$$\mathbf{H}_{init} = \begin{bmatrix} \mathbf{V}_g + \lambda\mathbf{A}(K \times N) \\ \mathbf{B}(D \times N) + \gamma\mathbf{1}(D \times N) \end{bmatrix} \tag{3.7}$$

$$\mathbf{W}_{init} = \begin{bmatrix} \mathbf{I}(K \times K) + \lambda\mathbf{O}(K \times K) & \mathbf{P}(K \times D) + \theta\mathbf{1}(K \times D) \\ \mathbf{Q}(F \times (D + K))) \end{bmatrix} \tag{3.8}$$

with $K = 50$ and $D = 25$ and with $\lambda = 1e^{-4}$, $\gamma = 0.1$ and $\theta = 0.2$. All entries in $\mathbf{A}, \mathbf{B}, \mathbf{O}, \mathbf{P}$ and $\mathbf{Q}$ are i.i.d samples from the uniform distribution $\mathcal{U}(0, 1)$ with boundaries $(0, 1)$. $\mathbf{I}$ is the identity matrix and $\mathbf{1}$ is a vector with all ones. Note that keeping $\mathbf{W}_g$ set to identity is suboptimal, since tuned label weights in $\mathbf{W}_g$ are helpful to model the duration of spoken keywords or to model word parts over multiple columns that combine to one keyword [see 35]. 100 iterations were used to find $\mathbf{H}$ and $\mathbf{W}$. Before each iteration the columns of $\mathbf{W}$ were normalised to sum to one. The parameters values were adopted from [12].

3.6.3 Phone HMM versus soft VQ Gaussians

Introduction

In the first experiment, we compare two processing flows that differ only in the mid-layer representation: soft VQ (section 3.5.3) or phone posteriorgrams (section 3.5.3).

Code books have been used before in NMF learning in numerous studies [11, 16, 22, 27, 36–39] and we set the baseline by adopting multiple parameter settings from these studies. As in [16], we used code books with different scales of granularity, $L = 20, 100$ and 400. Similar to [11], we used HAC's with frame-lags, $\tau = 2, 5$ and 9. Also the k-means procedure explained in section 3.5.3 was shared with former studies [11, 16, 22, 27, 36–39]. However, the soft VQ clusters are usually estimated by using all the data: seen and unseen data. Within the context of our study, this is regarded as an unrealistic simulation of the VUI training (see section 3.6.1). Nevertheless, we adopt this common (unrealistic) procedure to set a baseline in respect of earlier studies. This frequently applied soft VQ procedure is also an upper bound for the performance that can be expected from optimal training data in our VUI training simulation. The processing flow referring to this baseline setting is named MFCC_SVQ(ACN).

Code book training is completely unsupervised and data-driven, but the training of a phone HMM requires transcribed speech data. Spoken utterances of the user lack phonetic transcriptions, therefore, in a realistic VUI-user training scenario, phone models should be developed beforehand on transcribed speech data. We used WSJCAM0 for simulating the case that the phone models are trained on the native language of the end user, and CGN for the case that the trained phone models are originating from a different language. We refer to these processing flows with the names MFCC_PHN(WSJ) and MFCC_PHN(CGN), respectively. These two processing flows are considered realistic. Phone models trained on ACORNS are not considered realistic because the ACORNS speech data represents the spoken utterances of the user in our simulated VUI-user context. The spoken utterances of the user are unpredictable and lack phonetic transcriptions. However, similar to MFCC_SVQ(ACN), it is still an interesting case for investigating the potential gain if phone models could be trained without supervision. The processing scheme is called MFCC_PHN(ACN).

Results and discussion

In Figure 3.2, the average accuracies and standard error bars of the learning curves are depicted for each processing flow. For the baseline experiment MFCC_SVQ(ACN), a score of 98.5% is obtained for the largest training set

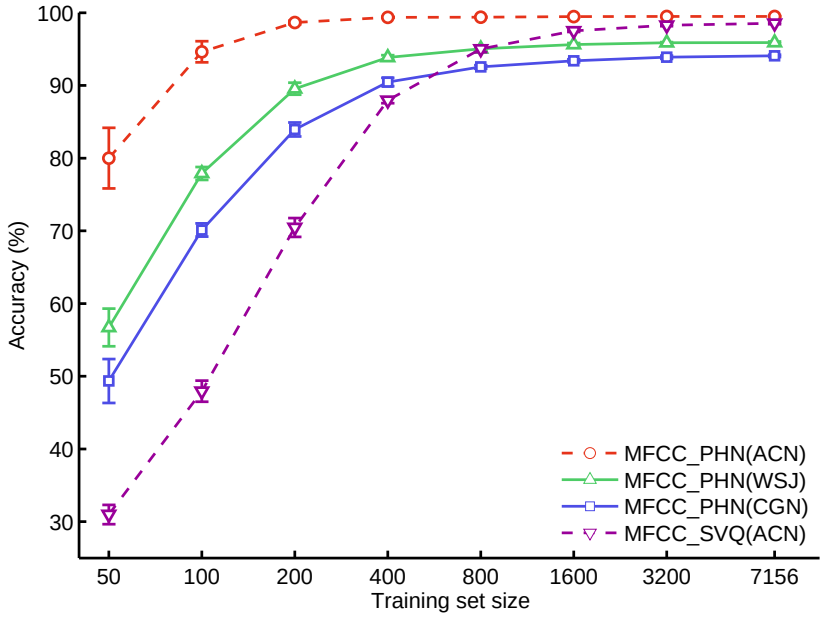


Figure 3.2: *Learning curves for processing flows using phone posteriorgrams or soft VQ as mid-level representation. Accuracy is plotted against the keyword-learning training set size. The error bars denote the standard error for the average accuracy over all folds and initializations.*

training set size	50	200	7156
MFCC_PHN(ACN)	80.0	98.7	99.5
MFCC_PHN(WSJ)	56.7	89.5	95,9
MFCC_PHN(CGN)	49.3	83.9	94.1
MFCC_SVQ(ACN)	31.0	70.5	98.5

Table 3.2: *Accuracies plotted in Figure 3.2 for keyword-learning training set sizes* $N = 50, 200$ *and* 7156.

(see Table 3.2). It is a score comparable to 98.1% obtained in similar conditions and presented in Table 4.5, page 97 in [12]. There, only one code book with $L = 500$ was used instead of multiple code books here, and the data was pooled over all 10 speakers instead of four speakers.

When we compare soft VQ against phone HMM, it is shown that the processing flows exploiting phone posteriorgrams have higher accuracies in the beginning of the learning curve compared to the soft VQ baseline. However, the flows MFCC_PHN(WSJ) and MFCC_PHN(CGN) level off earlier and accuracies are lower towards the end. In Table 3.2, asymptotic accuracies of 95.9% and 94.1% are obtained for the respective learning curves. Asymptotic accuracies are lower because the acoustic models are trained on speech data from different corpora to conform with a realistic VUI training scenario. Phones and clusters trained on ACORNS yield higher end scores but in a real training scenario, this training material would be unavailable prior to usage. The unrealistic training scenarios are plotted with dashed lines in Figure 3.2.

When we compare the processing flows with phones and clusters exploiting ACORNS training data, that is MFCC_PHN(ACN) and MFCC_SVQ(ACN) in Table 3.2, the Error Rate ($ER = 100\% - \text{accuracy}$) for $N = 50$ is 3.45 times lower for MFCC_PHN(ACN). A similar relative improvement with a factor of 2.86 is found between the same flows for $N = 7156$. Clearly, the choice of intermediate representations has a large influence on the learning speed and the asymptotic accuracy.

3.6.4 MIDA features versus MFCC features

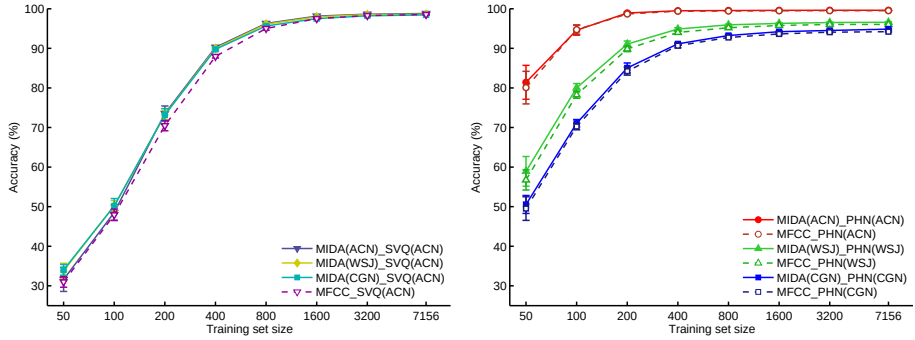
Introduction

In the low-layer representation of the architeture (see 3.1), two alternative spectro-temporal processing steps were explained. In the previous experiment, we used MFCC features in all processing flows, but here, we evaluate potential gains obtained from MIDA features in the same processing flows investigated before. We split the experiment in two parts: firstly, we evaluate the gains for MIDA using soft VQ, and secondly, we investigate MIDA features for the three phone recognisers adopted from the previous experiment.

In the first part, the three training sets for training the different MIDA transformations are ACORNS, WSJCAM0 and CGN. Code book training, the next step in the architecture, is then performed on the MIDA-transformed ACORNS features. According to the naming procedure (see section 3.6.2), the processing flows are called: MIDA(ACN)_SVQ(ACN), MIDA(WSJ)_SVQ(ACN) and MIDA(CGN)_SVQ(ACN). By keeping the code book training set constant, the effects of the three MIDA-transformation are comparable.

In the second part of the experiment, we implement the MIDA variant for each phone recogniser treated in the previous experiment resulting in the following

three processing flows MIDA(ACN)_PHN(ACN), MIDA(WSJ)_PHN(WSJ) and MIDA(CGN)_PHN(CGN).



(a) (b)

Figure 3.3: *Learning curves for processing flows comparing MIDA features against MFCC features for (a) soft VQ and (b) phone posteriorgrams. The error bars denote the standard error for the average accuracy.*

training set size	50	200	7156
MIDA(ACN)_SVQ(ACN)	34.0	73.5	98.8
MIDA(WSJ)_SVQ(ACN)	33.7	73.2	98.6
MIDA(CGN)_SVQ(ACN)	31.6	73.2	98.5
MFCC_SVQ(ACN)	31.0	70.5	98.5
MIDA(ACN)_PHN(ACN)	81.4	98.9	99.6
MFCC_PHN(ACN)	80.1	98.7	99.5
MIDA(WSJ)_PHN(WSJ)	58.9	91.1	96.6
MFCC_PHN(WSJ)	56.8	89.8	96.0
MIDA(CGN)_PHN(CGN)	50.6	85.1	94.9
MFCC_PHN(CGN)	49.5	84.2	94.2

Table 3.3: *Accuracies plotted in Figure 3.3 for keyword-learning training set sizes* $N = 50, 200$ *and* 7156.

Results and discussion

In Figure 3.3a and Table 3.3 it is shown that all learning curves based on MIDA features have higher or equal accuracies than the ones based on MFCC features.

However, most differences are non-significant.

In Figure 3.3b and Table 3.3 it is shown that all MIDA variants of the phone recognisers depicted in Figure 3.2 have higher accuracies over the whole range of the learning curves and some of these small differences are significant. The processing procedures investigated in the remaining experiments of this study are therefore all based on MIDA features.

The language of the training material influences the learning curves. When the supportive models are trained on a Dutch corpus (CGN), the scores are lower than when the models are trained on a British English corpus (WSJCAM). The best performance is obtained by using the same corpus for training both the acoustic and the keywords models, i.e. MIDA(ACN)_SVQ(ACN) and MIDA(ACN)_PHN(ACN).

3.6.5 User-centred keyword learning

Introduction

Contrary to the previous experiments (see section 3.6.3 and 3.6.4), user-centred NMF for keyword learning is pursued here with separate NMF keyword representations for every individual speaker. This contrasts the pooled keyword model counting for all four speakers together. Such a setup corresponds better to a realistic training context of the VUI where only a single end user trains and uses the system. We investigate the effect of speaker-specific input on keyword learning. The MIDA variant from the first experiment (section 3.6.3) was adopted here. Note that the learning curves share names with depicted learning curves in preceding experiments because the low- and the mid-layers are based on identical procedures. However, the obtained accuracies might differ as keyword learning is user-dependent. The setup is identical to the previous experiments (see section 3.6.2), except for the training sets containing utterances of each separate speaker only.

Results and discussion

The accuracies plotted in Figure 3.4 represent the score averaged over all four speakers. The error bars reflect the standard deviation of the scores of the four speakers.

The same qualitative differences as the ones observed in Figure 3.2 can be observed in Figure 3.4, but all accuracies are considerably higher. For instance, similar to Figure 3.2, phone posteriorgrams outperform the soft

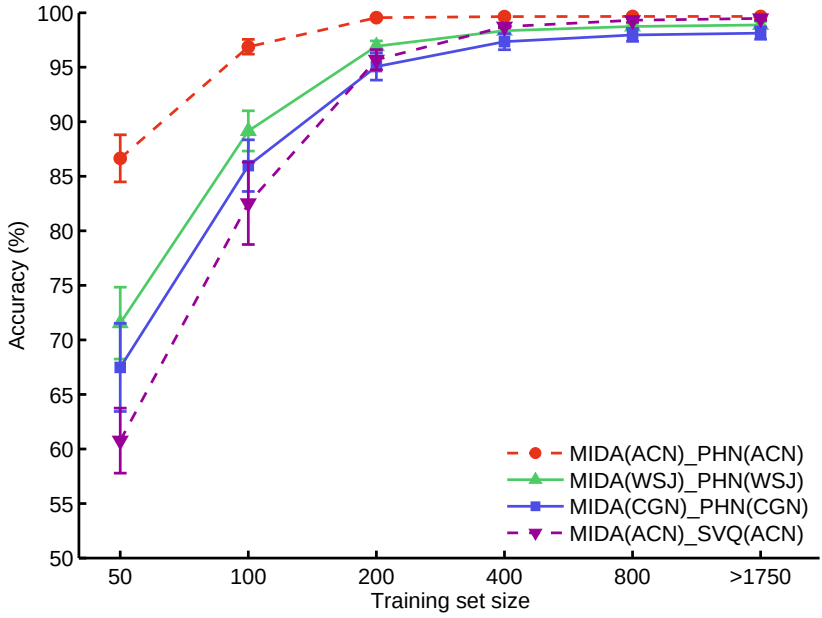


Figure 3.4: *Learning curves for user-centred keyword learning. The error bars denote the standard error for the mean accuracy of the four speakers.*

training set size	50	200	>1750
MIDA(ACN)_PHN(ACN)	86.7	99.5	99.7
MIDA(WSJ)_PHN(WSJ)	71.5	96.9	98.9
MIDA(CGN)_PHN(CGN)	67.5	95.1	98.1
MIDA(ACN)_SVQ(ACN)	60.8	95.6	99.5

Table 3.4: *Accuracies plotted in Figure 3.4 for keyword-learning training set sizes* $N = 50, 200$ *and* > 1750.

VQ representation in the beginning of the learning curve, but the soft VQ representation outperforms two out of the tree streams based on phone posteriorgrams at the end of the learning curve. Exact accuracies are given in Table 3.4.

Accuracies are higher compared to speaker-pooled keyword learning because NMF models only need to take into account the vocalizations of a single speaker

instead of all four speakers together. Discriminative representations are easier to build when the words are spoken by a single user. Despite the fact that the largest training set is four times larger for the speaker-pooled folds in the preceding experiments, the shorter learning curves here finish with higher accuracies. The highest accuracy, here, for all curves based on a realistic VUI training, is 98.9% for MIDA(WSJ)_PHN(WSJ) (see Figure 3.4), but 96.6% for the same flow in the previous experiment (see Figure 3.3b). By making the VUI more personalised, higher accuracy is yielded than with the initial baseline (yielding 98.5% in Table 3.2). Thus a personalised system obtains better accuracy for a realistic training scenario than the baseline system actually exploiting oracle training data for codebook training. This baseline has been applied in many former studies [11, 16, 22, 27, 36–39].

3.6.6 User-centred code book training

Introduction

The advantage of training code books beforehand is that large speech corpora can be used, such as those employed in the field of speech recognition. However, the acoustic-model training set is then recorded in different conditions (e.g. different microphones, different room acoustics and maybe cleaner speech) and with different speakers than the speech data originating from the user. We use WSJCAM0 as acoustic-model training set to simulate the case where the acoustic-model training set is different from the keyword-training set ACORNS. We refer to this processing flow with the name MIDA(WSJ)_SVQ(WSJ).

The speech data of the user has no phonetic transcriptions in a real VUI-usage environment, but, since code book training is data-driven, the user data can be used to train the code books. However, the data will be limited to the set of utterances that the user has spoken up until a particular moment in time, and thus, the training data is rather scarce especially during the initial VUI usage. We refer to the processing flow as MIDA(WSJ)_SVQ(ACN,SSD). We follow the code book training procedure explained in section 3.5.3 but we add one constraint by prohibiting further splitting of clusters when the number of frames joining one cluster becomes less than 78 frames, a measure that allows for a more reliable estimation of the covariance matrix of the Gaussians. However, by fulfilling this constraint, the number of clusters is variable and gradually increases with the number of utterances in the training sets. For instance, for the training set sizes with $N = 50, 100, 200$, 800, and > 1750, we obtained on average code book sizes of $L = 51, 93, 148, 191$ and 330 for all the folds.

The aim of this experiment is to investigate whether code book training for scarcely available but speaker-dependent data yields higher accuracies compared

to the case where data is speaker-independent, but abundantly available in the field of speech recognition. These two realistic cases are accompanied by one unrealistic case where code books are trained on all available speaker-dependent data in ACORNS. It serves as a reference for the case that large amounts of speech data from the user would be available before the VUI usage. We call the learning curve MIDA(WSJ)_SVQ(ACN,SD). Note that such a scenario can be realistic when speech from the end user is recorded beforehand for example by reading a standard text before the usage of the VUI.

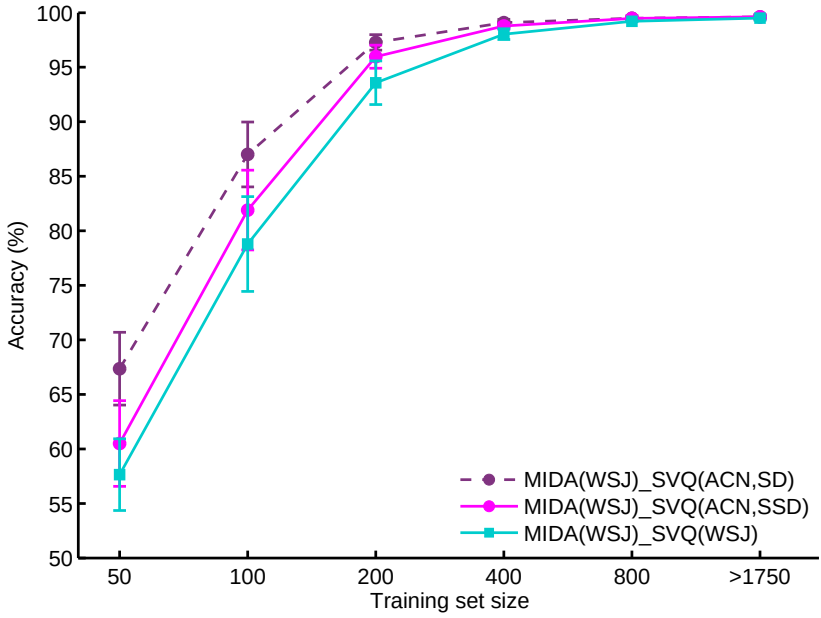


Figure 3.5: *Learning curves for user-centred keyword learning and speaker-(in)dependent code book training. The error bars denote the standard error for the mean accuracy of the four speakers.*

training set size	50	200	>1750
MIDA(WSJ)_SVQ(ACN,SD)	67.4	97.3	99.6
MIDA(WSJ)_SVQ(ACN,SSD)	60.5	95.8	99.6
MIDA(WSJ)_SVQ(WSJ)	57.6	93.6	99.5

Table 3.5: *Accuracies plotted in Figure 3.5 for keyword-learning training set sizes* $N = 50, 200$ *and* > 1750.

Results and discussion

When the two realistic learning curves are compared with each other (solid lines in Figure 3.5), the best performances is obtained for MIDA(WSJ)_SVQ(ACN,SSD) over the whole range of the learning curve. Better performance was expected when plenty of speaker-dependent data is available allowing for better matching code books. However, small-sized code books matching the speaker's vocalizations also result in better scores in the beginning of the learning curve. In different words, small datasets matching the speaker's vocalization are preferable to many hours of speech data recorded in different conditions with different speakers and different vocabularies leading to more phonetic variation. User-centred soft VQ methods are also attractive for deviant speech for the reason that code books will give a good match to the end user.

For the smallest training set size, the MIDA(WSJ)_SVQ(ACN,SSD) is 7% behind in absolute accuracy compared to the unrealistic best-case scenario MIDA(WSJ)_SVQ(ACN,SD) (see Table 3.5). The difference represents the potential gain that can be achieved hypothetically, if pre-recorded speech of the end user would be available beforehand.

3.6.7 Stream combination

Introduction

In this experiment, the goal is to combine the realistic processing flows that yielded the best results for the average user in all the former experiments. Within the set of realistic learning curves, MIDA(WSJ)_SVQ(ACN,SSD) provided the highest accuracy at the end of the learning curve and MIDA(WSJ)_PHN(WSJ) provided the highest accuracy in the beginning of the learning curve. By combining both processing flows, we investigate whether the best of both worlds can be obtained over the whole range of the learning curve.

The two flows are combined in NMF by stacking the data matrices $\mathbf{V}_a$ of both processing flows in one large data matrix giving both streams equal weights, i.e. both streams are normalised so the sum of all entries in each stream are equal. Naturally, weights can be tuned to favour one of the two performance indicators.

The two streams MIDA(WSJ)_PHN(WSJ) and MIDA(WSJ)_SVQ(ACN,SSD) are adopted from the former two experiments in section 3.6.5 and 3.6.6. The combined stream is called MIDA(WSJ)_comb.

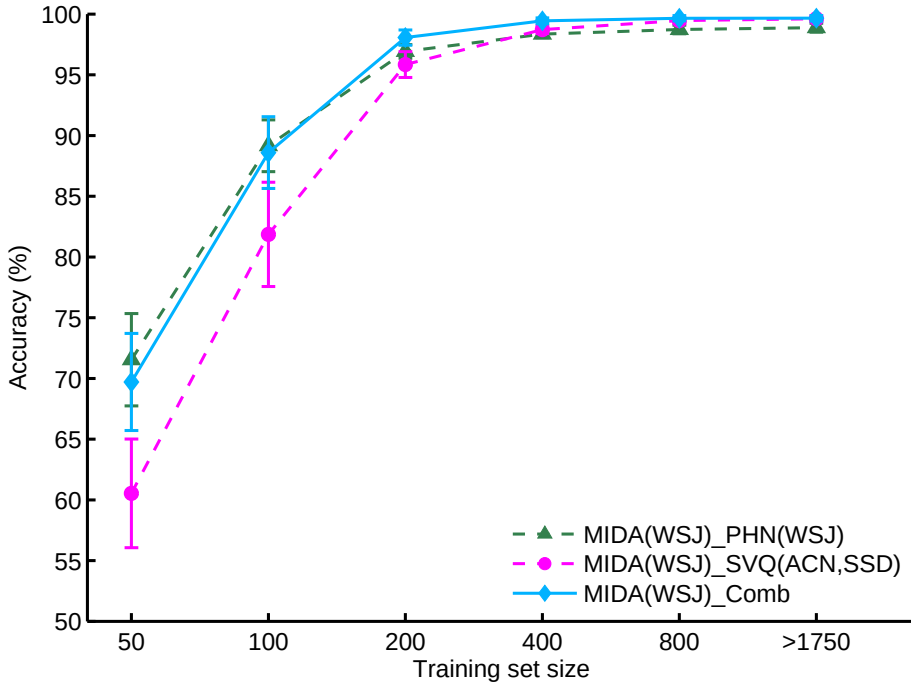


Figure 3.6: *Learning curves for the combination of two realistic processing flows adopted from the previous studies. The error bars denote the standard error for the mean accuracy of the four speakers.*

training set size	50	200	>1750
MIDA(WSJ)_comb	69.7	98.0	99.7
MIDA(WSJ)_PHN(WSJ)	71.5	96.9	98.9
MIDA(WSJ)_SVQ(ACN,SSD)	60.5	95.8	99.6

Table 3.6: *Accuracies plotted in Figure 3.6 for keyword-learning training set sizes* $N = 50, 200$ *and* > 1750.

Results and discussion

When training set sizes are larger or equal to 200 examples, higher accuracies are obtained for MIDA(WSJ)_comb compared to its constituents: MIDA(WSJ)_PHN(WSJ) and MIDA(WSJ)_SVQ(ACN,SSD) (see Figure 3.6

and Table 3.6). For training set sizes $N = 50$ and $N = 100$, the combined processing flow performs slightly worse than the best one of its constituent streams but still performs much better than the worst one of its constituent streams. For the largest training set size, the accuracies of the combined and the best of its constituent streams are similar. The combined processing scheme seems to approximates the best scores of its constituent counterparts and demonstrate therefore better overall performances.

3.7 General discussion

We investigated NMF-based performance in a series of experiments simulating the realistic training conditions of the VUI-user context. For instance, the environmental local conditions of the user, such as room acoustics and the vocabulary spoken by the user are not known beforehand. Likewise, the data used for training phone HMM or MIDA transformations, like WSJCAM0 or CGN, are recorded in different conditions with different speakers and vocabularies than the data used for simulating the VUI training. We progressed to adapted models by first using user-centred NMF training (section 3.6.5) and secondly by using user-centred code book training (section 3.6.6). Both steps improved the performance to a great extent.

We took more measures on the way to fast learning. In the first experiment (section 3.6.3), we introduced phone posteriorgrams to enhance the feature vectors in NMF and we obtained better performance than the more commonly used soft VQ features [12, 22]. Also the use of MIDA features in the second experiment (section 3.6.4) allowed for a slight improvement in performance.

The optimal performance is obtained by combining a phone recogniser trained on WSJCAM0 initiating a head start and the use of user-centred speaker and set-size dependent code book training allowing for high asymptotic accuracies of the learning curve at the end (see section 3.6.7). Both processing streams are considered realistic scenarios in the VUI usage context.

The user group consists of people with limb impairments for which voice control contributes to their independence of living. The majority of the user group is expected to have normal intelligibility, but some physical impairments are caused by neuromuscular diseases, therefore disarthric speech is expected too. The combined stream demonstrates promising results. One stream using phone posteriorgrams allows fast word learning for normal spoken utterances and one stream based on more basal soft VQ features allows to build up new word representations from scratch. The second stream is particularly interesting for people with a speech impairment. In future research, we will investigate

whether the vocal user interface is able to anticipate dysarthric speech. Some preliminary research in that respect has been carried out in [40].

3.7.1 Posteriorgrams as feature vectors

Developmental studies demonstrate that humans build an intermediate representation of speech sounds in function of semantic content [41, 42]. We show that machine learning of the semantic content of signals is largely improved when a mid-level representation is built based on speech-sound categories like phones or clusters (see also [43]). Especially, the use of posteriorgrams to enhance feature vectors seems to be a promising procedure in NMF learning. For example, in [16] we used hard VQ instead of soft VQ for the initial baseline in the first experiment, namely MFCC_SVQ(ACORNS), and obtained a score of 32.6%, 48.2% and 95.6% for training sets in ACORNS of size 100, 200 and 9821 utterances. Here we obtained 47.9%, 70.5% and 98.5% for exactly the same conditions using the posteriorgram version of hard VQ, namely soft VQ.

The processing flows based on the posteriorgram of a pretrained phone recogniser are especially efficient in the beginning of the learning curve. Phones were modelled by a tri-state HMM expressing phones as variable sequences of frame-based acoustic observations. The generative HMM models can cope with many forms of spectro-temporal variation and in that sense, their structure incorporates a great deal of information on human speech in general by extracting information from large annotated corpora beforehand. In the meantime, they consist of very compact feature representations of the data at hand, facilitating the search of latent recurrent keyword patterns and allowing for fast word learning rates in NMF.

Conversely, the Gaussian models used in soft VQ are less complex, therefore limiting the training data required to estimate the parameters. Since a sufficient number of code words are required to accurately represent speech for recognition purposes, feature vectors based on soft VQ are less compact and more training examples for NMF keyword learning are required. However, code book training is data-driven, affording the pursuit of code books on-the-fly, leading to representative clusters regarding the speech of the user. The more user-specific clusters allow for better performance in the long run, especially for users with deviant or dysarthric speech. The positive results of the user-centred approach in training demonstrates the potential asset by grounding the learning process in the environment of the user.

Future research entails the evaluation of simple data-driven phone-like subword models [15, 44] embodying the best of both worlds: user-centred acoustic models similar to soft VQ and generative HMM models allowing a compact feature

representation. The challenge is to furnish the acoustic models with a limited set of parameters allowing accurate discrimination: a limited set in order to achieve fast learning, but discriminative in order to obtain high accuracies in the long run. An alternative is to create a model with a growing number of parameters evolving to a more and more complex model as more data becomes available. Finally, the third possibility and the one pursued here is to combine different procedures with different strengths. In the last experiment (section 3.6.7) we combined two processing flows at the front-end of NMF using equal weights for both streams. Future research entails finding optimal weights for different input streams based on the work of [28] and the dynamical adaptation of weights in function of the learning curve. Since Phone posteriorgrams and soft VQ level off at different instants, it is likely that optimal weights are changing during the learning process.

3.7.2 Related work on fast learning

The exact experimental evaluation of our results with others is out of the scope of this study as it is difficult to compare results when they are based on different databases, procedures and scoring. For instance, we tested the feasibility of our approach and limited ourselves to a realistic learning scenario. The performance indicator accuracy is not complementary to the more common used Word Error Rate (WER) in the sense that the accuracy does not comprise correct keyword order in an utterance but only the proportion of correctly detected keywords in utterances with one to four keywords embedded in it. On the other hand, the categorical complexity of our task consists of 50 keywords and 30 filler words while a database, like for instance TIDIGITS, contains a lexicon of eleven words and a corpus like WSJCAM0 contains a lexicon of $64,000$ words. Moreover, the supervision in our task is weak in the sense that it does not comprise word order or segmentation. Therefore, the comparison of our approach with related work is rather qualitative.

Fast vocabulary acquisition has also been investigated in an HMM architecture. In [45], the lexicon (9 digits, “oh” and “zero”) from the TIDIGITS database was learned from just a few training examples with supervision. In their framework, optimal parameters were first sought for the initialization of the HMM by a multiple sequence alignment procedure in which an initial ergodic HMM is transformed into multiple left-right HMM’s, one for each word. They used a large margin classifier to obtain good generalization to new instances as the classifier was trained on a few examples. For continuous speech, they obtained an average word error rate (WER) of 13.7% after three learning examples and 1.7% after 10 learning examples. The models of [45] were speaker-independent and their learning procedure was incremental.

Fast learning in an NMF framework has been investigated by [14]. They pursued a computational model for the discovery of new words by young infants. In the task at hand, a vocabulary of 13 keywords embedded in a carrier sentence was learned and it was asserted that 20 to 25 learning examples per word are sufficient to approximate a recognition accuracy at asymptotic level. The same NMF procedure and data subset was pursued in [15] to learn a new vocabulary of 10 extra keywords after the acquisition of a vocabulary of 40 keywords and some filler words. They used self-discovering HMM subword units to enhance the acoustic input and they achieved similar acquisition rates to the ones presented in this study. Their investigation was rather aimed at the adaptation capacity of a fully trained NMF model to newly encountered words. Our work builds further on former NMF studies. By using phone posteriorgrams and more user-centred acoustic and keyword models (section 3.6.7), we obtained an average ER of 26% after three learning examples and an ER of 1.5% after 10 learning examples of a keyword. Our method has an advantage over other approaches because we use a pretrained phone recogniser trained on annotated databases.

In [16], it was found that accuracies mainly depend on the number of correct examples per keyword and not on the number of utterances in the training set. If an average command consists of two keywords, for instance an object name and an action, then 2.25 correct demonstrations per command are needed on average to obtain a keyword recognition rate above 90%. Five correct demonstrations on average allow to reach asymptotic levels. We think that the average user will experience this training effort as a reachable goal with rewarding return. In that sense, fast learning in a realistic setting —which was the aim of our study —is achieved for normal speech.

3.7.3 Conclusion

We aim at designing a VUI that learns to understand normal and ultimately deviant speech by associating spoken commands to actions on a device during its usage. The VUI is trained by the end user by mining the speech input and the changes that are provoked on a device. The real learning process will take place in the environment of the user but it is simulated in our experiments in a realistic manner as a machine learning problem grounded with keyword labels, i.e. labels that specify the action on a device. We focussed on fast learning and high asymptotic accuracy of the learning curve.

Simple commands consisting of two keywords, like “Switch on the lights, please”, can be learned by five demonstrations. Fast learning in a realistic setting —which was the aim of our study —was therefore achieved and we demonstrated fast learning by taking several measures on the way: phone posteriorgrams were introduced in the first experiment, MIDA features were pursued in the second

experiment and user-centred NMF for keyword learning improved performance in the third experiment. In the fourth experiment, the results were in favour of user-centred code books trained on scarce data instead of using massive amounts of data from different speakers. Finally, for the combined processing flow in the fifth experiment, we obtained an accuracy in keyword detection of 99.7% (starting from 98.5% for the baseline) and we improved the accuracy for the smallest training set from 30.9% using state-of-the-art NMF approaches to 69.7%, that is a reduction in error rate of more than a factor two. Additionally, we focussed on realistic training scenarios to have a sense on how such a system would perform in a real-life training scenario as grounding of the VUI training in the user's environment is the most important key aspect of the self-learning VUI.

A.0 Overview processing streams

Table A.0: *The naming convention for different processing flows with respect to the low- and mid-layer data preparation for NMF-based keyword learning. Only processing flows used in the experiments are depicted. Italic formatted names indicate processing flows which are regarded as unrealistic because they make use of unavailable user-specific data to train the acoustic models. "SD" refers to speaker-dependent training and "SDD" refers to speaker and set-size dependent training.*

			Low-layer		
			MFCC features	MIDA features	
		Training corpus	no training	ACORNS	WSJCAM0	CGN
Mid-layer	Soft VQ	ACORNS	*MFCC_SVQ(ACN)*	*MIDA(ACN)_SVQ(ACN)*	*MIDA(WSJ)_SVQ(ACN)*	*MIDA(CGN)_SVQ(ACN)*
		WSJCAM0			MIDA(WSJ)_SVQ(WSJ)	
		CGN				
		ACORNS, SD			*MIDA(WSJ)_SVQ(ACN,SD)*	
		ACORNS, SSD			MIDA(WSJ)_SVQ(ACN,SSD)	
	Phones	ACORNS	*MFCC_PHN(ACN)*	*MIDA(ACN)_PHN(ACN)*		
		WSJCAM0	MFCC_PHN(WSJ)		MIDA(WSJ)_PHN(WSJ)	
		CGN	MFCC_PHN(CGN)			MIDA(CGN)_PHN(CGN)

3.1 References

[1] T. Paek and D. Chickering, "Improving command and control speech recognition on mobile devices: using predictive user models for language modeling," *User modeling and user-adapted interaction*, vol. 17, no. 1, pp. 93–117, 2007. pages 61

[2] T. Heinroth, M. Grotz, F. Nothdurft, and W. Minker, "Adaptive speech understanding for intuitive model-based spoken dialogues," in *Proc. LREC*, pp. 1281–1288, 2012. pages 61

[3] A. Potamianos and S. Narayanan, "Spoken dialog systems for children," in *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, vol. 1, pp. 197–200, IEEE, 1998. pages 61

[4] R. Kuhn, J.-C. Junqua, P. Nguyen, and N. Niedzielski, "Rapid speaker adaptation in eigenvoice space," *Speech and Audio Processing, IEEE Transactions on*, vol. 8, no. 6, pp. 695–707, 2000. pages 61

[5] M. Parker, S. Cunningham, P. Enderby, M. Hawley, and P. Green, "Automatic speech recognition and training for severely dysarthric users of assistive technology: the stardust project," *Clinical linguistics & phonetics*, vol. 20, no. 2-3, pp. 149–156, 2006. pages 62

[6] H. H. Clark and E. F. Schaefer, "Contributing to discourse," *Cognitive science*, vol. 13, no. 2, pp. 259–294, 1989. pages 62

[7] J. van de Loo, J. F. Gemmeke, G. De Pauw, J. Driesen, H. Van hamme, and W. Daelemans, "Towards a self-learning assistive vocal interface: Vocabulary and grammar learning," in *Proc. of the workshop Speech and Multimodal Interaction in Assistive Environments (SMIAE)*, 2012. pages 62

[8] J. Gemmeke, B. Ons, M. Tessema, J. Van de Loo, G. De Pauw, W. Daelemans, J. Huyghe, J. Derboven, L. Vuegen, B. Van Den Broeck, and H. Van hamme, "Self-taught assistive vocal interfaces: An overview of the aladin project," in *Proceedings of Interspeech*, 2013. pages 62

[9] J. C. Caicedo, J. BenAbdallah, F. A. González, and O. Nasraoui, "Multimodal representation, indexing, automated annotation and retrieval of image collections via non-negative matrix factorization," *Neurocomputing*, vol. 76, no. 1, pp. 50–60, 2012. pages 62

[10] Z. Akata, C. Thurau, and C. Bauckhage, "Non-negative matrix factorization in multimodality data for segmentation and label prediction," in *16th Computer Vision Winter Workshop*, 2011. pages 62

[11] J. Driesen, J. Gemmeke, and H. Van hamme, "Weakly supervised keyword learning using sparse representations of speech," in *Proc. ICASSP*, (Kyoto, Japan), pp. 5145–5148, 2012. pages 62, 64, 68, 69, 70, 71, 77, 83

[12] J. Driesen, *Discovering words in speech using matrix factorization.* PhD thesis, K.U.Leuven, ESAT, July 2012. pages 63, 68, 76, 78, 87

[13] M. Sun, *Constrained Non-negative Matrix Factorization for Vocabulary Acquisition from Continuous Speech.* PhD thesis, K.U.Leuven, ESAT, 2012. pages 63

[14] L. ten Bosch, J. Driesen, H. Van hamme, and L. Boves, "On a computational model for language acquisition: modeling cross-speaker generalisation," in *Text, Speech and Dialogue*, pp. 315–322, Springer, 2009. pages 63, 64, 90

[15] J. Driesen and H. Van hamme, "Fast word acquisition in an NMF-based learning framework," in *Proc. ICASSP*, (Kyoto, Japan), pp. 5137–5140, 2012. pages 88, 90

[16] B. Ons, J. F. Gemmeke, and H. Van hamme, "Label noise robustness and learning speed in a self-learning vocal user interface," in *Proc. of the International Workshop on Spoken Dialog Systems (IWSDS)*, (Ermenonville, France), 2012. pages 63, 70, 77, 83, 88, 90

[17] L. Boves, L. ten Bosch, and R. Moore, "Acorns-towards computational modeling of communication and recognition skills," in *Proc. IEEE int. Conf. On Cognitive informatics*, (California, USA), pp. 349–355, 2007. pages 63, 73

[18] W. Quine, *Word and object*, vol. 4. MIT press, 1964. pages 64

[19] H. Van hamme, "Hac-models: a novel approach to continuous speech recognition," in *Proc. Interspeech*, (Brisbane, Australia), pp. 255–258, 2008. pages 64, 69, 70, 71

[20] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *In IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980. pages 66

[21] K. Demuynck, *Extracting, Modelling and Combining Information in Speech Recognition.* PhD thesis, K.U.Leuven, ESAT, February 2001. pages 68

[22] M. Sun and H. Van hamme, "A two-layer non-negative matrix factorization model for vocabulary discovery," in *In ICML'11 Symposium on Machine Learning in Speech and Language Processing*, (Bellevue, Washington, USA), 2011. pages 68, 77, 83, 87

[23] F. Wessel, R. Schluter, K. Macherey, and H. Ney, "Confidence measures for large vocabulary continuous speech recognition," *Speech and Audio Processing, IEEE Transactions on*, vol. 9, no. 3, pp. 288–298, 2001. pages 69

[24] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989. pages 69

[25] M. Van Segbroeck and H. Van hamme, "Unsupervised learning of time-frequency patches as a noise-robust representation of speech," *Speech Communication*, vol. 51, pp. 1124–1138, 2009. pages 69

[26] J. Driesen and H. Van hamme, "Modelling vocabulary acquisition, adaptation and generalization in infants using adaptive bayesian plsa," *Neurocomputing*, vol. 74, no. 11, pp. 1874–1882, 2011. pages 70

[27] J. Driesen, J. F. Gemmeke, and H. Van hamme, "Data-driven speech representations for NMF-based word learning," in *Proceedings of the workshop on Statistical and Perceptual Audition*, (Portland, OR, USA), 2012. pages 70, 77, 83

[28] J. Driesen and H. Van hamme, "Supervised input space scaling for non-negative matrix factorization," *Signal Processing*, vol. 92, no. 8, pp. 1864–1874, 2012. pages 70, 89

[29] D. Lee and H. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999. pages 71

[30] V. Stouten, K. Demuynck, and H. Van hamme, "Discovering phone patterns in spoken utterances by non-negative matrix factorization," *IEEE Signal Processing Letters*, vol. 15, pp. 131–133, 2008. pages 72

[31] T. Altosaar, L. ten Bosch, G. Aimetti, C. Koniaris, K. Demuynck, and H. van den Heuvel, "A speech corpus for modeling language acquisition: Caregiver," in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)* (N. C. C. Chair), K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, and D. Tapias, eds.), (Valletta, Malta), European Language Resources Association (ELRA), may 2010. pages 73

[32] T. Robinson, J. Fransen, D. Pye, J. Foote, and S. Renals, "Wsjcam0: A british english speech corpus for large vocabulary continuous speech recognition," in *Proc. ICASSP*, (Detroit, Michigan, USA), 1995. pages 74

[33] N. Oostdijk, "The spoken dutch corpus. overview and first evaluation.," in *Proc. LREC*, (Genoa, Italy), 2000. pages 74

[34] K. Demuynck, J. Roelens, D. Van Compernolle, and P. Wambacq, "Spraak: An open source speech recognition and automatic annotation kit," in *Proc. International Conference on Spoken Language Processing*, (Brisbane, Australia), 2008. pages 75

[35] B. Ons, J. F. Gemmeke, and H. Van hamme, "NMF-based keyword learning from scarce data," in *Automatic Speech Recognition and Understanding Workshop, ASRU*, (Olomouc, Czech Republic), 2013. pages 76

[36] J. Driesen, L. ten Bosch, and H. Van hamme, "Adaptive non-negative matrix factorization in a computational model of language acquisition," in *Proc. Interspeech*, (Brighton, UK), pp. 1711–1714, 2009. pages 77, 83

[37] J. Driesen and H. Van hamme, "Modelling vocabulary acquisition, adaptation, and generalization in infants using adaptive bayesian plsa," *Neurocomputing*, vol. 74, pp. 1874–1882, 2011. pages

[38] M. Sun and H. Van hamme, "Image pattern discovery by using the spatial closeness of visual code words," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pp. 205–208, IEEE, 2011. pages

[39] M. Sun and H. Van hamme, "Tri-factorization learning of sub-word units with application to vocabulary acquisition," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 5177–5180, IEEE, 2012. pages 77, 83

[40] B. Ons, N. Tessema, J. van de Loo, J. F. Gemmeke, G. De Pauw, W. Daelemans, and H. Van hamme, "A self learning vocal interface for speech-impaired users," *Proceedings SLPAT 2013*, pp. 1–9, 2013. pages 88

[41] K. Miyawaki, J. Jenkins, W. Strange, A. Liberman, R. Verbrugge, and O. Fujimura, "An effect of linguistic experience: The discrimination of [r] and [l] by native speakers of japanese and english," *Attention, Perception, & Psychophysics*, vol. 18, no. 5, pp. 331–340, 1975. pages 88

[42] J. Werker and C. Lalonde, "Cross-language speech perception: Initial capabilities and developmental change.," *Developmental psychology*, vol. 24, no. 5, p. 672, 1988. pages 88

[43] N. H. Feldman, T. L. Griffiths, S. Goldwater, and J. L. Morgan, "A role for the developing lexicon in phonetic category acquisition.," *Psychological review*, vol. 120, no. 4, p. 751, 2013. pages 88

[44] M. Sun and H. Van hamme, "Joint training of non-negative tucker decomposition and discrete density hidden markov models," *Computer Speech & Language*, vol. 27, no. 4, pp. 969–988, 2013. pages 88

[45] I. A. Clemente, M. Heckmann, and B. Wrede, "Incremental word learning: Efficient hmm initialization and large margin discriminative adaptation," *Speech Communication*, vol. 54, no. 9, pp. 1029–1048, 2012. pages 89

Chapter 4

Model adaptations for scarce data

This chapter is based on the following article:
B., Ons, J.F., Gemmeke, H., Van hamme, "NMF-based keyword learning from scarce data," *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU),* pp. 392-397, (Olomouc, Czech Republic), 2013.

4.1 Abstract

This research is situated in a project aimed at the development of a vocal user interface (VUI) that learns to understand its users specifically persons with a speech impairment. The vocal interface adapts to the speech of the user by learning the vocabulary from interaction examples. Word learning is implemented through weakly supervised non-negative matrix factorization (NMF). The goal of this study is to investigate how we can improve word learning when the number of interaction examples is low. We demonstrate two approaches to train NMF models on scarce data: 1) training word models using smoothed training data, and 2) training word models that strictly correspond to the grounding information. We found that both approaches can substantially improve word learning from scarce training data.

4.2 Context and contributions of the chapter

In the preceding chapter, we developed a procedure to acquire speaker-dependent acoustic clusters. This approach is based on batch learning of the training examples that emerge during usage. The number of clusters increases as more training examples are collected. During the development of the experimental design in **chapter 3**, we found an effect on the performance by manipulating the number of clusters: large codebooks led to poorer results than consise codebooks if the number of training examples is low. This finding led to the reflection that the NMF-based method was susceptible to overfitting.

The contribution of this chapter is that we introduced two measures to deal with overfitting. The most effective method was 'smoothing'; the other measure, referred to as 'restricted word learning', is less effective but interesting because it can be conceived as a typical cognitivist approach. In 'restricted word learning', direct association between acoustic features and semantic labels are created by tying (restricting) the entries in the incidence matrix ($\mathbf{H}$) to the semantic supervised labels. A label is then a symbol for which an acoustic representation is sought. The evaluation of the NMF approach with more self-organising freedom (a characteristic of emergentism) against its more restricted symbol-representation variant (a characteristic of cognitivism) is an interesting contribution of this chapter.

4.3 Introduction

Command and Control (C&C) speech recognition allows users to control different conditions in their environment like the central heating or the light units in the house, but also the interaction with devices like smartphones or computers. This study is situated in the "Adaptation and Learning for Assistive Domestic Vocal Interfaces" (ALADIN) project [1, 2] aimed at the development of a Vocal User Interface (VUI) that can understand normal as well as deviant speech. The VUI learns to understand the user who is able to choose his own words, phrases or sounds in order to control the target application at hand (see **chapter 6**).

We meet this objective by grounding the word learning process of the VUI in the environment of the end user, so that the VUI is trained by mining the speech input from the end user and the changes that are provoked on a device. For instance, the user says: "Please, turn on the television" and turns on the television with the remote control. The learning problem is a machine learning problem where the user has to demonstrate the intended action to the VUI, and by doing so, he provides supervision to the spoken utterance [2]. The supervision for training the speech recognizer is only weak, since the changes provoked on the device, resulting for instance from a button push, cannot be transformed in an orthographic transcription with correct word order as is required in training conventional automatic speech recognition systems based on Hidden Markov Models (HMMs) [3].

As an alternative, non-negative matrix factorization (NMF) has been presented as a useful machine learning procedure to discover and learn the acoustic representation of spoken words guided by weak supervision [4–6]. In short, NMF decomposes utterance-based representations into two low-rank representations, one representing the recurring acoustic patterns such as spoken words, and one describing which recurring patterns are active in each utterance.

The goal of this study is to investigate how we can improve fast vocabulary acquisition in the state-of-the-art NMF approach [4]. Fast learning is an essential objective as it reduces the user's effort to train the system and allows faster workability of the VUI. This is achieved when word models trained on *scarce* speech data can still generalize to new speech signals. We propose two approaches to improve the word recognition rates: 1) *smoothing* of the acoustic model posterior probabilities in order to avoid over-training of the NMF word models, and 2) *restricting* the acoustic representation of the word models to correspond exactly to the supervision data, i.e. the grounding information. If a spoken word and its supervision information only appear one time, it is not a recurrent pattern and difficult to detect by NMF. By imposing the supervision, we essentially seek representations for words that appeared only once. We will

evaluate the effectiveness of both approaches by doing word learning experiments with increasing amounts of training data.

The chapter is organized as follows. In Section 4.4, we briefly explain *supervised NMF* and the processing steps to build the feature vectors for NMF, namely *soft vector quantisation* (soft VQ) [7] and the *histograms of acoustic co-occurrence (HAC)* [6]. In Section 4.5, we describe the two approaches to improve the generalization of the models. We conduct two experiments, one for each method and we report the results in section 4.6. Finally, in Section 4.7 and 4.8 we discuss the proposed methods and conclude with our conclusions and thoughts on future work.

4.4 Background

4.4.1 Acoustic representation

Fixed length feature vectors are required for NMF. We build utterance-based fixed length vectors by transforming the acoustic feature vectors into a Gaussian posteriorgram [7] and by accumulating the posterior probabilities to an histogram of acoustic co-occurrence (HAC) [4, 6] .

A posteriorgram is a two dimensional data structure containing the posterior probability that a frame-based feature vector (first dimensions: time) was emitted by a particular acoustic unit (second dimension: class). Here, the classes are Gaussians obtained by k-means clustering followed by the estimation of a full covariance Gaussian based on all frame observations falling in each respective cluster [7, 8]. Each entry in the posteriorgram is the relative (normalized) likelihood that the observation was emitted from the respective Gaussian.

The posteriorgram of an utterance has a variable length that depends on the number of frames in an utterance. We create HAC features to build a fixed-length vector for each utterance and to incorporate time information. In the HAC, the probability of co-occurrence between frames, τ frames apart from each other, is accumulated over one whole utterance for all possible cluster pairs. Coarser and more fine grained code books as well as more time information are added by stacking HAC's with different time lags and different codebooks in one utterance-based vector. The vector length F depends on te number L_i of Gaussians in each codebook $i = 1, \ldots, C$ and the number of time lags T: $F = T \times \sum_{i=1}^{C} L_i^2$. The data matrix consisting of the acoustic representation of N utterances is denoted by $\mathbf{V}_a(F \times N)$ with F the number of features.

4.4.2 Grounding information

In addition to the acoustic representation, there is a second input stream providing utterance-based supervision denoted by $\mathbf{V}_g(K \times N)$ with K the number of words defining the demonstrated actions on a device, also called keywords. Supervision in each utterance is indicated as follows: there is one row in $\mathbf{V}_g$ for each keyword and its entries represent the number of times that the respective word was uttered in the n^{th} utterance. In the context of the VUI of Section 4.3, this assumes VUI actions such as pushing a button are related to one or more keywords. Supervision is weak in the sense that the absence or presence of keywords are indicated without any chronological information within the utterance.

4.4.3 The supervised NMF framework

Training

NMF [9] decomposes a data matrix $\mathbf{V}$ into the product of two lower rank matrices, $\mathbf{W}$ and $\mathbf{H}$. A variant to NMF is supervised NMF [4, 6] where supervision $\mathbf{V}_g$ is added to the data matrix $\mathbf{V}_a$. This grounding includes an additional part in the lower rank matrix $\mathbf{W}$, namely $\mathbf{W}_g$ to regularize the factorization in correspondence with the supervision. The model is:

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_g \\ \\ \mathbf{V}_a \end{bmatrix} \approx \begin{bmatrix} \mathbf{W}_g \\ \\ \mathbf{W}_a \end{bmatrix} \mathbf{H} = \mathbf{W}\mathbf{H} \tag{4.1}$$

with all entries in $\mathbf{V}$, $\mathbf{W}$ and $\mathbf{H}$ constrained to be non-negative.

The purpose of supervised NMF learning is to uncover the acoustic representation of each keyword. The columns in $\mathbf{W}_a$ represent the latent structure (recurring patterns) of the columns in $\mathbf{V}_a$ associated to a keyword (expressed in $\mathbf{W}_g$). The columns in $\mathbf{H}$ indicate which patterns are combined to approximate the columns in $\mathbf{V}$.

When the total set of vocal commands contains K keywords, $\mathbf{W}$ should count at least K columns, but in practice, some D extra columns are added to $\mathbf{W}$ to model the filler words.

Different loss functions are possible and the most appropriate loss function depends on the statistical structure of the data matrix. An appropriate loss function for the approximation in Eq. 4.1 assuming that entries in $\mathbf{V}$ are counts

of events is the generalised Kullback-Leibler divergence (gkld) or I-divergence:

$$D_{KL}(\mathbf{V}||\mathbf{WH}) = \sum_i \sum_n \left[v_{in} \log \frac{v_{in}}{[\mathbf{WH}]_{in}} - v_{in} + (\mathbf{WH})_{in} \right] \quad (4.2)$$

with $i = 1, \ldots, I$, $I = K + F$ and $n = 1, \ldots, N$.

Lee and Seung [9] derived the following alternating multiplicative update rules for minimizing Eq. 4.2 as a function of the entries h_{rn} of $\mathbf{H}$ and w_{ir} of $\mathbf{W}$. Convergence is guaranteed to a local optimum:

$$h_{rn} \leftarrow h_{rn} \frac{\sum_i \frac{v_{in}}{[\mathbf{WH}]_{in}} w_{ir}}{\sum_q w_{qr}} \quad (4.3)$$

$$w_{ir} \leftarrow w_{ir} \frac{\sum_n \frac{v_{in}}{[\mathbf{WH}]_{in}} h_{rn}}{\sum_p h_{rp}} \quad (4.4)$$

with v_{in} entries of $\mathbf{V}$, and $r = 1, \ldots, R = K + D$ with R the inner dimension of $\mathbf{W}$ and $\mathbf{H}$. After each update of $\mathbf{W}$, we normalise its columns to sum to unity in order to prevent arbitrary scaling of $\mathbf{W}$ and $\mathbf{H}$.

In supervised NMF, the first K rows in $\mathbf{H}$ are initialized as $\mathbf{V}_g$ and the first $K \times K$ entries in $\mathbf{W}_g$ are initialized as the identity matrix [8]. A small random number is added to $\mathbf{W}_g$. The initialization procedure helps convergence to a solution with keyword representations in the first K columns of $\mathbf{W}_a$. All entries in $\mathbf{W}_a$ are randomly initialized.

The solutions for $\mathbf{H}$, $\mathbf{W}_a$ and $\mathbf{W}_g$ obtained by update rules in Eq. 4.3 and 4.4 are denoted by $\mathbf{H}^*$, $\mathbf{W}_a^*$ and $\mathbf{W}_g^*$.

Recognition

Keyword recognition is tested on a separate set of new utterances denoted by $\mathbf{V}_t$. $\mathbf{H}_t^*$ is found by minimizing the generalized Kullback-Leibler divergence between $\mathbf{V}_t$ and $(\mathbf{W}_a^*\mathbf{H}_t)$ with known $\mathbf{W}_a^*$:

$$\mathbf{H}_t^* = \arg\min_{\mathbf{H}_t} D_{KL}(\mathbf{V}_t||\mathbf{W}_a^*\mathbf{H}_t) \quad (4.5)$$

The optimization problem in Eq. 4.5 is a convex problem as $\mathbf{W}_a^*$ is held fixed, and the solution $\mathbf{H}_t^*$ is used to provide the keyword activation matrix $\mathbf{A}$ as follows:

$$\mathbf{A} = \mathbf{W}_g^*\mathbf{H}_t^* \quad (4.6)$$

The higher the score in the rows of $\mathbf{A}$, the more likely that the respective keyword has appeared in the spoken test utterances.

4.5 Proposed methods

We propose two methods to improve the word learning from scarce training data: learning from *smoothed* data and restricting the optimalization procedure to follow the supervision, referred to as *restricted word learning*.

4.5.1 Smoothing

In this method, we propose to smooth the data matrix by imposing smoothness on the posteriorgrams. The smoothed posteriorgram with entries $\hat{P}_{t_i,\theta}$, with θ denoting a Gaussian from the set Φ of Gaussians and t_i the timestamp of the respective frame, smoothing is described as follows:

$$\hat{P}_{t_i,\theta} = \frac{(P_{t_i,\theta})^{\zeta}}{\sum_{\theta \in \Phi} (P_{t_i,\theta})^{\zeta}} \tag{4.7}$$

with the exponent $0 < \zeta < 1$ leading to smoother (flatter) posterior probabilities.

We investigated the effect of smoothing for small and large training sets using two different smoothing conditions: we smoothed the training data and the test data in the first condition while we only smoothed the training data, but not the test data, in the second condition. If smoothing is helpful in reducing noise and irrelevant small-scale features, we expect an improvement in both cases over all training set sizes. The improvements gained by smoothing are then essentially depending on the resolution of the data. A second procedure is to smooth the training data but not the test data. This causes a mismatch between training data and test data and should degrade accuracies. However, if better performance is also obtained by smoothing the training data but not the test data for small data sets, a strong indication is provided that the smoothing of scarce data is able to provide word models that generalize better to new instances, thus overcome overfitting.

4.5.2 Restricted word learning

In this method, we keep the first K rows of $\mathbf{H}$ fixed during the multiplicative optimization updates (see Eq. 4.3 and 4.4). The first K rows of $\mathbf{H}$ correspond to the supervision (The first K rows of $\mathbf{V}_g$) and indicate the occurrence of a keyword by a number 0 or 1. However, keeping the entries in $\mathbf{H}$ fixed to 0 or 1 is actually suboptimal as a value different from 1 allows us to model the duration of the spoken words. Longer words are spread over more acoustic frames, and therefore, they have larger acoustic co-occurence counts. Since the keyword

representations modelled by the first K colums of $\mathbf{W}$ are all normalized to sum to unity, the differences in word length can only be reflected in the entries of $\mathbf{H}$. Nevertheless, the tying of $\mathbf{H}$ to the supervision data in $\mathbf{V}_g$ is a good initial approximation of the optimal solution to $\mathbf{H}$ and keeping these values fixed to this initial approximation is not going to harm the optimization process too much while we gain by reducing the dimensionality of the optimization problem. We therefore expect better word models by restricting the optimization of $\mathbf{H}$ for small data sets but not for large data sets. We call this approach "restricted word learning".

4.6 Experiments

4.6.1 Introduction

We evaluate potential gains for the use of smoothing and restricted word learning as explained in Section 4.5. In the first experiment, we implemented seven smoothing values for ζ in Eq. 4.7, $\zeta = 0.025$, 0.05, 0.1, 0.2, 0.4, 0.6 and 1. The baseline is given by a value of 1-smoothing. We implemented the two smoothing conditions explained in Section 4.5.1 and evaluated smoothing for small and large training sets, $N = 50, 100, 200$ and 1785.

In the second experiment, we investigated six training set sizes, $N = 50, 100, 200, 400, 800$ and 1785 utterances against two different multiplicative update schemes. In the baseline condition, we used the traditional update rules as expressed in Eq. 4.3 and 4.4. In the restricted condition, we used a different update rule for $\mathbf{H}$ as explained in Section 4.5.2. The performance is evaluated by the accuracy expressed as the percentage of correct recognized keywords.

4.6.2 Experimental setup

Speech material

To mimic a usage situation in which no speech material of a user is (yet) available when training the word learning system, code book training is carried out on a different database than the one used for keyword learning. This means the low-level acoustic model is speaker-independent and the recording conditions differ from the user environment. We used the "Wall Street Journal corpus recorded at the University of Cambridge, phase 0", WSJCAM0 [10] for this purpose, which is the UK English equivalent of a subset of the US English Wall street Journal corpus (WSJ0).

For keyword learning we used the UK English subset of the ACORNS corpus [11] developed in the second year of the ACORNS project and we selected the four speakers with the most recorded utterances. The test sets counted $593, 594, 596$ and 599 utterances for the four respective speakers and we utilized training sets of increasing sizes with ultimately, $N = 1790, 1786, 1789$ or 1791 utterances for the largest training set for the four speakers, respectively. In ACORNS, utterances consist of 1 to 4 different keywords embedded in a carrier sentence with unrelated filler words. In total, there are 50 unique predefined keywords and 30 filler words. The choice for the corpus fit well for the purpose of evaluating the performance of the VUI since the supervision is weak (a bag of words) and the size and complexity of the data is similar to a common home automation task.

Features

Feature extraction was done by using Mutual Information Discriminant Analysis or MIDA [12], MIDA features consist of a linear combination of 22 log-MEL spectral dimensions and their first and second order differences (Δ and $\Delta\Delta$). The linear combination is aimed at maximizing the mutual information between the MIDA features and phone classes. The MIDA transformation was learned using the corpus WSJCAM0.

We used three code books of dimension $L = 20, 100$ and 400. Each code word corresponded to one Gaussian, and posteriorgrams were created using the procedure described in Section 4.4.1.

HAC representations were created as explained in Section 4.4.1 using three frame lags, $\tau = 2, 5$ and 9. For each combination of frame lag and code book, there is one posteriorgram per utterance. For each utterance we obtained one fixed-length vector with the dimensionality determined by the number of code books, their sizes and the number of frame lags: $F = 3 \times (20^2 + 100^2 + 400^2) = 511200$ features for each utterance, however, feature vectors are very *sparse*.

Implementation

In addition to the initialisation procedure explained in Section 4.4.3, $D = 25$ was chosen for both experiments. Preliminary experiments showed that 100 iterations are sufficient to reach convergence. We applied five different initialisations of each respective combination of smoothing, training set size, speaker and update scheme. One possible problem could be that for $N \leq K + D$ (i.e., the training set size $N = 50$), the rank $\mathbf{W}$ is larger than the rank of $\mathbf{V}$.

However, the non-negativity constraints and the supervision in NMF inhibit a trivial solution.

4.6.3 Results

The resulting accuracies are shown in Figure 4.1 and 4.2. For each method, there is a graph showing the average keyword recognition accuracy as a function of smoothing, see 4.1, and as a function of set size for restricted and unrestricted word learning, see 4.2. The error bars denote the 95% confidence interval after controlling the variation due to the speaker variability using the procedure described in [13].

Smoothing

We found significant improvements with respect to the baseline (horizontal dotted lines in 4.1) for almost all levels of smoothing (solid lines in 4.1) when using small training sets, with $N = 50$, $N = 100$ and $N = 200$. In the smallest training set, $N = 50$, every keyword was spoken at least one time. We were able to obtain a baseline accuracy of 57% and improved the result to 66% by smoothing, an improvement of 8% with respect to the baseline. However, we did not find a significant improvement for the same smoothings in the largest training set $N > 1785$.

Smoothing the training set but not the test set (dashed lines in 4.1) shows a tendency to improve the accuracy even more for the smallest data set $N = 50$. This is a remarkable result given the mismatch in resolution between training and test data. Probably, smoothing training data allows for models that generalise better while no smoothing during decoding prevents information loss. This finding suggests that NMF is susceptible to overfitting and that Bayesian approaches to NMF should be considered (see Chapter 6) However, the opposite trend is seen for the largest training set size. The dashed line depicting the accuracy for the largest training set in Figure 4.1 declines for more smoothing.

Clearly, smoothing behaves differently for small and large data sets and smoothing improves accuracy for scarce training data.

Restricted word learning

For small data sets, $N = 50$ and $N = 100$, we found a significant improvement in accuracy with respect to the baseline (the dashed lines in Figure 4.2) by restricting word learning (the solid lines in Figure 4.2) as explained in

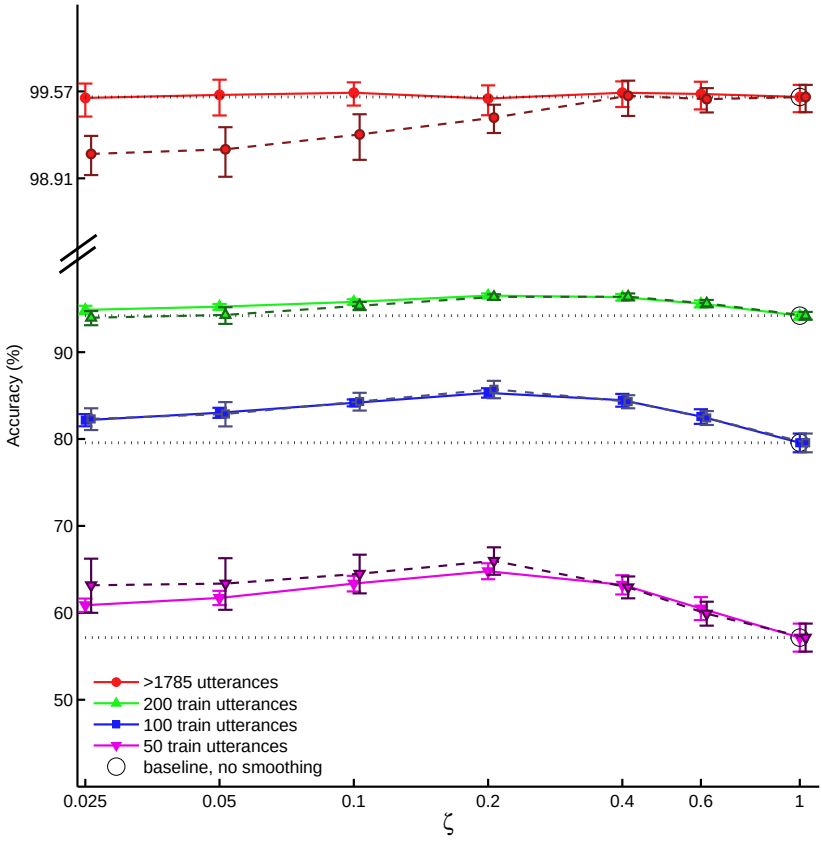


Figure 4.1: *Smoothings for different training set sizes. The error bars denote the 95% confidence intervals. The dashed lines are accuracies against different smoothing values used to smooth the training data whereas the test data was not smoothed. The solid lines are accuracies against smoothing values used to smooth both sets, training and test data. The horizontal dotted lines indicate the respective baseline performance (no smoothing).*

Section 4.5.2. However, for larger data sets, ($N \geq 200$), the opposite effect is displayed favoring common optimization update rules as expressed in Eq. 4.3 and 4.4. For the smallest training set, counting 50 utterances, we have an average baseline accuracy of 57% and we obtained an accuracy of 66% by restricted word learning, an improvement of 8%, quite similar to the effect of smoothing. However, for the largest dataset, $N > 1785$, the baseline (see Figure 4.2) gave an accuracy of 99.5% while restricted word models led to a

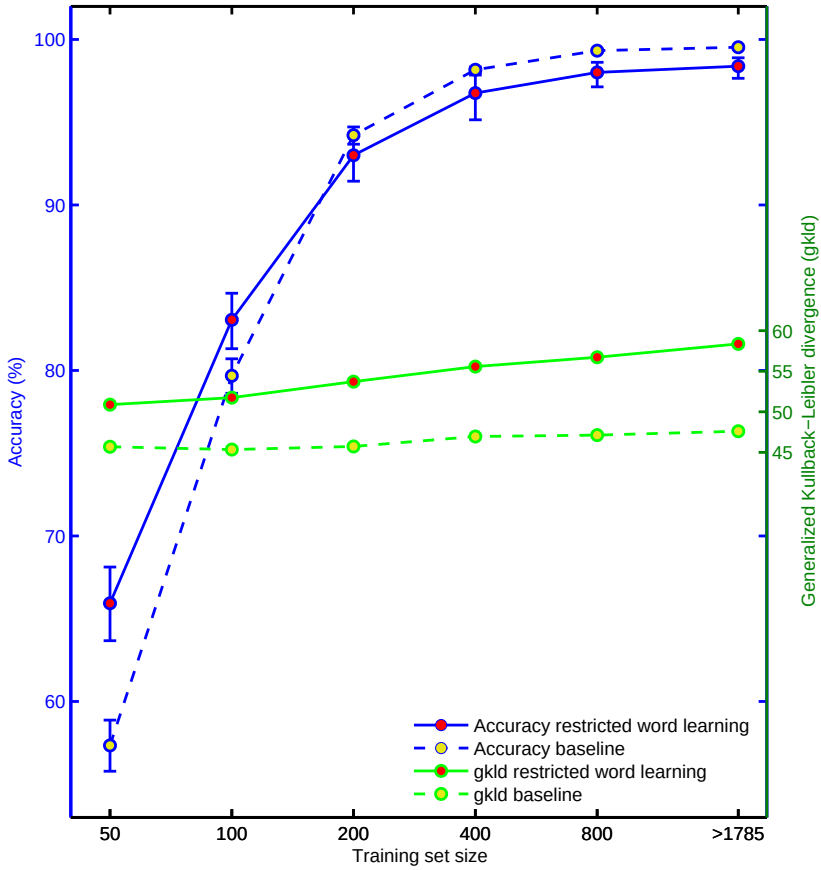


Figure 4.2: *Restricted word learning. The blue lines are the accuracies on the left y-axis against different training set sizes using common NMF updates (the dashed line) or restricted word learning (the solid line). The green lines depict the generalized Kullback-Leibler divergence (gkld, see Eq. 4.2) on the right Y-axis, between the predicted occurence of words in the test set and the plain truth.*

lower accuracy of 98.7%. Restricted word models are only helpful in the case of scarce data. The cost function of the test set, i.e. of $\mathbf{A}$ in Eq. 4.6 are depicted in a green color in Figure 4.2. Although restricted word learning allows for a better prediction of word occurence in the test set, these predictions have a higher generalized Kullback-Leibler divergence compared with the baseline.

4.7 Discussion

Clearly, there is a relation between the amount of training data that is available and the improvements by either smoothing and restricted word learning. In general, both techniques are helpful if the number of training examples is low. The different effects of both techniques on small and large training sets demonstrate that optimization issues and model training pose a distinct challenge for scarce data. To the best of our knowledge, this distinction has not been given a lot of attention in the literature.

4.7.1 Smoothing

Smoothing appears to be effective for all but the largest data sets. Moreover, the optimal parameter value for smoothing is independent of the size of the training set. This can be understood as follows. Smoothing the probabilities of acoustic events causes more overlap of the Gaussians in the feature space. Without smoothing, only one or two Gaussians contribute significantly to the total probability mass of an observation as most observations lie close to the centre of a single high-dimensional Gaussian. The effect of smoothing is that observations are described by multiple Gaussians and a larger mass in their posteriorgram is shared if they are located in the same region of the feature space. As the shared probability mass between different observations corresponding to the same keyword label increases, it becomes easier to detect a recurrent pattern in the case of scarce data.

If training and test sets are smoothed, smoothing also increases the robustness of the feature representations. The training-test mismatch makes their position in the feature space uncertain within some neighbourhood. A small shift in position will affect the non-smoothed posteriorgram much more than the smoothed posteriorgram. Smoothing therefore reduces the noise level of the observation at the cost of some fine-scale resolution. A coarser but more robust representation is especially helpful for the case of scarce training data. However, for large training sets, when test sets are not smoothed, a coarser resolution of the training set affects the performance negatively as the training-test mismatch becomes larger.

A third positive effect of smoothing is related to the use of the KLD divergence. Probabilities which are underestimated during training on scarce data may have a detrimental effect during testing because of the singularities at zero and the asymmetry of the KLD. Such features have an unreasonably large impact on the total value of the cost function. The use of smoothing increases those probabilities and generally balances the impact of the acoustic features.

4.7.2 Restricted word models

By imposing a solution in favour of the supervision introduced in $\mathbf{H}$ (see Section 4.4.3), we find an adequat representation of the keywords for which supervision is provided, i.e. the first K columns in $\mathbf{W}$, but it raises questions about the inadequate representation of the filler words for which no supervision is available, the D remaining columns in $\mathbf{W}$. The presence of filler words is randomly initialized in $\mathbf{H}$ and unsupervised learning of the filler words is solely based on detecting recurrent acoustic patterns. If filler words are adequatly represented, they are helpful for keyword recognition because they separate irrelevant patterns from relevant keyword patterns in the utterance-based representation (a bag of features). This does raise the question of whether *any* number of garbage columns ($D > 0$) can be beneficial for scarce training data, but this is left as future work.

Although better results are obtained for restricted word learning if the number of training examples is low, these better results are accompanied with a higher divergence. In different words, normal update rules learn better to minimize the generalised Kullback-Leibler divergence than the proposed approach, but keyword recognition accuracy is lower for scarce data. This observation suggests that modifications to the objective function taking into account the availability of the training data and the mathematical expression of the supervision could lead to better solutions.

4.8 Conclusion

We demonstrated two techniques, smoothing and restricted word learning, to improve weakly supervised NMF learning on scarce training data. Smoothing was shown to be an effective method to substantially accelerate word learning on small data sets while maintaining the good accuracies on larger training sets. These findings are valuable since they showed that optimization issues and model training pose a distinct challenge if the availability of data is limited. Moreover, the second technique, restricted word learning seemed to improve the performance only for scarce data sets. This result demonstrates that self-organizing freedom (as opposed to constraint) is beneficial for word learning when exploiting sufficient data resources. The self-organization of the VUI is a typical attribute of the emergentist view on machine learing.

4.9 References

[1] J. van de Loo, J. F. Gemmeke, G. De Pauw, J. Driesen, H. Van hamme, and W. Daelemans, "Towards a self-learning assistive vocal interface: Vocabulary and grammar learning," in *Proc. of the workshop Speech and Multimodal Interaction in Assistive Environments (SMIAE)*, 2012. pages 101

[2] J. F. Gemmeke, J. van de Loo, G. De Pauw, J. Driesen, H. Van hamme, and W. Daelemans, "A self-learning assistive vocal interface based on vocabulary learning and grammar induction," in *Proc. INTERSPEECH*, pp. 1–4, 2012. pages 101

[3] I. A. Clemente, M. Heckmann, and B. Wrede, "Incremental word learning: Efficient hmm initialization and large margin discriminative adaptation," *Speech Communication*, vol. 54, no. 9, pp. 1029–1048, 2012. pages 101

[4] J. Driesen, J. Gemmeke, and H. Van hamme, "Weakly supervised keyword learning using sparse representations of speech," in *Proc. ICASSP*, (Kyoto, Japan), pp. 5145–5148, 2012. pages 101, 102, 103

[5] L. ten Bosch, J. Driesen, H. Van hamme, and L. Boves, "On a computational model for language acquisition: modeling cross-speaker generalisation," in *Text, Speech and Dialogue*, pp. 315–322, Springer, 2009. pages

[6] H. Van hamme, "Hac-models: a novel approach to continuous speech recognition," in *Proc. Interspeech*, (Brisbane, Australia), pp. 255–258, 2008. pages 101, 102, 103

[7] M. Sun and H. Van hamme, "A two-layer non-negative matrix factorization model for vocabulary discovery," in *In ICML'11 Symposium on Machine Learning in Speech and Language Processing*, (Bellevue, Washington, USA), 2011. pages 102

[8] J. Driesen, *Discovering words in speech using matrix factorization.* PhD thesis, K.U.Leuven, ESAT, July 2012. pages 102, 104

[9] D. Seung and L. Lee, "Algorithms for non-negative matrix factorization," *Advances in neural information processing systems*, vol. 13, pp. 556–562, 2001. pages 103, 104

[10] T. Robinson, J. Fransen, D. Pye, J. Foote, and S. Renals, "Wsjcam0: A british english speech corpus for large vocabulary continuous speech recognition," in *Proc. ICASSP*, (Detroit, Michigan, USA), 1995. pages 106

[11] L. Boves, L. ten Bosch, and R. Moore, "Acorns-towards computational modeling of communication and recognition skills," in *Proc. IEEE int. Conf. On Cognitive informatics*, (California, USA), pp. 349–355, 2007. pages 107

[12] K. Demuynck, *Extracting, Modelling and Combining Information in Speech Recognition*. PhD thesis, K.U.Leuven, ESAT, February 2001. pages 107

[13] D. Cousineau, "Confidence intervals in within-subject designs: A simpler solution to loftus and masson's method," *Tutorial in Quantitative Methods for Psychology*, vol. 1, no. 1, pp. 42–45, 2005. pages 108

Chapter 5

The self-taught vocal interface for dysarthric speech

This chapter is based on the following article:
B., Ons, J.F., Gemmeke, H., Van hamme, "The self-taught vocal interface," *EURASIP journal on Audio, Speech and Music Processing,* 2014, 2014:43, doi:10.1186/s13636-014-0043-4 `http://asmp.eurasipjournals.com/content/2014/1/43`.

5.1 Abstract

Speech technology is firmly rooted in daily life, most notably in command-and-control (C&C) applications. C&C usability degrades quickly, however, when used by people with non-standard speech. We pursue a fully adaptive Vocal User Interface (VUI) which can learn both vocabulary and grammar directly from interaction examples, achieving robustness against non-standard speech by building up models from scratch. This approach raises feasibility concerns on the amount of training material required to yield an acceptable recognition accuracy. In previous work we proposed a VUI based on non-negative matrix factorization (NMF) to find recurrent acoustic and semantic patterns comprising spoken commands and device-specific actions, and showed its effectiveness on unimpaired speech. In this work, we evaluate the feasibility of a self-taught VUI on a new database called DOMOTICA-3, which contains dysarthric speech with typical commands in a home automation setting. Additionally, we compare our NMF-based system with a system based on Gaussian mixtures. The evaluation favours our NMF-based approach, yielding feasible recognition accuracies for people with dysarthric speech after a few learning examples. Finally, we propose the use of a multi-layered semantic frame structure and demonstrate its effectiveness in boosting overall performance.

5.2 Context and contributions of the chapter

At the time of experiments in this chapter, the ALADIN corpus was recorded and contained typical command-and-control data. The first contribution is the evaluation of the VUI model on dysarthric speech. A second contribution is that we investigated different semantic structures. Typical applications with an embedded ASR component proceed via an intermediate step: acoustics are linked to words and phrases via the ASR component and these words are used as symbolic input to the semantic processes or functions in the application at hand. In the ALADIN approach, acoustics are related to semantics without the intermediate translation of acoustics into words. Therefore, the semantic organization influences the performance of the VUI to a great extent. Unfortunately, to the best of our knowledge, there is no generic approach for building the most convenient semantic structure that applies to each VUI implementation. With the availability of two home-made corpora in the ALADIN project, some experience is gained with the implementation of the ALADIN approach for typical targeted applications. Examining different semantic structures allowed us to develop a sense for the influence of the different semantic structures. The third contribution is that we demonstrate that a semantic structure with more hierarchical layers affects accuracy positively. In

a final contribution, we investigate the interdepence of semantic values. The activation of semantic values is correlated. For example, the co-activation of <switch on> and <light> is more likely than the co-activation of <switch on> and <door>. A final contribution is that we constructed a decision process in which we collected and propagated all piecewise activations through a hierarchical frame structure. This collection of activations allows a global view on the active frame.

5.3 Introduction

Currently, modern voice control technology is available in many applications, such as direct voice input (DVI) in aviation [1], information requests using Siri and speech driven home automation. Command and Control (C&C) appliances afford hands-free control, thus enhancing the independence of the physically incapacitated. Unfortunately, speech commands are sometimes misinterpreted when words overstep lexical boundaries and word sequences do not fit the preset grammars. Moreover, C&C- appliances frequently fail to interpret dialectic or impaired speech, often encountered with physically challenged people. Consequently, people with non-standard speech are increasingly excluded from the growing market of voice driven applications. The goal of this work is to investigate a Vocal User Interface (VUI) model which is able to learn words and grammars from end users; improving accessibility of C&C applications.

Over the past decade, various approaches have been proposed to improve the usability of Automatic Speech Recognition (ASR) for speech-impaired users. For example, in [2–4], speaker-independent acoustic models were adapted to speaker-dependent and speaker-adapted models; both providing better recognition of user-specific vocalizations. Besides adaptation, dysarthric speakers also improved the recognition likelihood of their words by training the consistency of their pronunciations ([5–7]); thus users can adapt their vocalizations in order to alleviate the ASR shortcoming to cope with severe vocal variability. In [8, 9], the increased phonetic variability associated with dysarthric speech was addressed by a system enabling more suitable HMM topologies for each phoneme in the speaker's repertoire. Another example is [10], where user-needs were surveyed and reflected in the design of a VUI for which an isolated word recognition system with a customizable command list and a built-in word prediction function was proposed to improve usability of typical services on mobiles and tablets. Although these approaches resulted in considerable improvements in usability, the accessibility of voice control technology still needs to widen to cater for users with non-standard or impaired speech (see [11, 12]).

State of the art ASR is typically based on HMM acoustic models developed with Gaussian mixture (GMM) continuous emission densities and context-dependent bi- or triphones models with multiple states per model. These language-dependent models are trained on hundreds of thousands of recorded and annotated speech utterances. Some applications in voice-enabled automated home environments use ASR models together with a speaker adaptation procedure to improve ASR performance for specific users or user groups. For example, the DIRHA [13], SWEET-HOME [14] and HomeService [15] projects aim for voice-enabled assistive technology in home environments for people with a physical impairment. In the DIRHA and SWEET-HOME project, maximum likelihood linear regression (MLLR) speaker adaptation is used starting from a speaker-independent ASR system. In the HomeService project, speaker-independent ASR models were obtained using normal or dysarthric speech followed by maximum a posteriori (MAP) speaker-adaptation. These approaches require annotated language-dependent speech material in addition to annotated user-specific speech material. The advantage of the adaptation approach is that the amount of user-specific speech material composes only a fraction of the data required for building a speaker-dependent state of the art speech recognizer. Speaker-dependent data usually requires an enrolment session and automated or non-automated transcriptive resources. Contrary to the adaptation approach, the basic approach here and in the (ALADIN) project, see [16] for an overview, is to build a VUI model that starts from scratch and learns from speech and demonstrations of the end user without transcription. Considering the VUI usability, the training procedure requires the ability to learn from a few examples and should be able to work with easily obtainable annotations such as content or context information. In our language-independent approach, the VUI learns to understand spoken commands by mining the speech input from the end user and the changes that are provoked on a device.

The first aim of the study is to test the feasibility of the learning procedure to construct speech patterns such as words from a few examples and content-related annotations. The speech of the user and the content information entered by the device are two sources of information that we combine by using Non-negative Matrix Factorisation (NMF, see [17]). This procedure allows the VUI to learn co-occurring patterns from two information sources. In [18], we proposed a novel grammar induction technique based on HMM learning and semantic descriptions of commands guiding the learning process. Here, we propose multi-layered semantic structures and implement the semantic dependencies in a tree structure. The second aim of the study is to compare the new semantic structure with the ones employed in [18]. For this, we use two databases; one with recordings of normally speaking subjects playing a card game by voice, and another one with commands provoked in a virtual home automation setting for

people diagnosed with dysarthria. The first database is referred to as PATCOR, whereas the second one is a new database called DOMOTICA-3. Besides the validation of new semantic structures, we will evaluate the NMF procedure as well by comparing our NMF-based framework with a Gaussian Mixture Model (GMM)-based baseline system.

The remainder of the chapter is organised as follows. In section 5.4, we describe the learning framework, including the semantic and acoustic representations as well as the NMF learning procedure. In section 5.5, we describe a reference model employing GMMs instead of NMF. We proceed by describing the databases used for evaluation in section 5.6. Subsequently, we explain the semantic structure of spoken commands (cf. section 5.7) before conducting a series of experiments (cf. section 5.8) where we evaluate the feasibility of our approach and the effectiveness of more layered semantic structure. We present our conclusion and thoughts on future work in the final section 5.9.

5.4 Language learning in the vocal user interface

A schematic overview of the learning framework is depicted in Figure 5.1. Here, two different types of data are processed; one processing stream is depicted in the upper part and builds up a *semantic representation*, while the other one, depicted in the lower part, builds up an *acoustic representation*. In the upper processing stream, device-specific functionality is parsed into a semantic *frame description*. The conversion is guided by a hand-crafted *semantic frame structure* as indicated with the dotted arrow pointing towards the arrow leading to the block "frame description" (cf. section "5.4.1"). The frame description is turned into a *label vector* and passed on to the *NMF* module.

In the lower part of Figure 5.1 (cf. section 5.4.2), spectro-temporal features are extracted and transformed into Mel-Frequency Cepstral Coefficients (MFCC's, cf. section 5.4.2). The MFCC features are converted into a *posteriorgram* and the horizontal dotted arrow from the right leaving from the block *Codebook/Gaussians*, indicates that, for this, intermediate procedures like *codebook* training and clustering are needed (cf. section 5.4.2). The posteriorgram is then converted into an utterance-based representation by using Histograms of Acoustic Co-occurrence (*HAC*, cf. section 5.4.2) after which the NMF training takes place. The depicted matrices denoted by $\mathbf{H}$ contain column-wise entries for each learning example representing loads on recurrent patterns in the data matrices $\mathbf{V}_s$ and $\mathbf{V}_a$, which are represented by the columns in the depicted matrices $\mathbf{W}_s$ and $\mathbf{W}_a$, with the subindex connoting the semantic or the acoustic stream respectively. The large bi-directed arrow between the two matrices $\mathbf{H}$, indicates that a common matrix is sought for $\mathbf{H}$, thus common loads

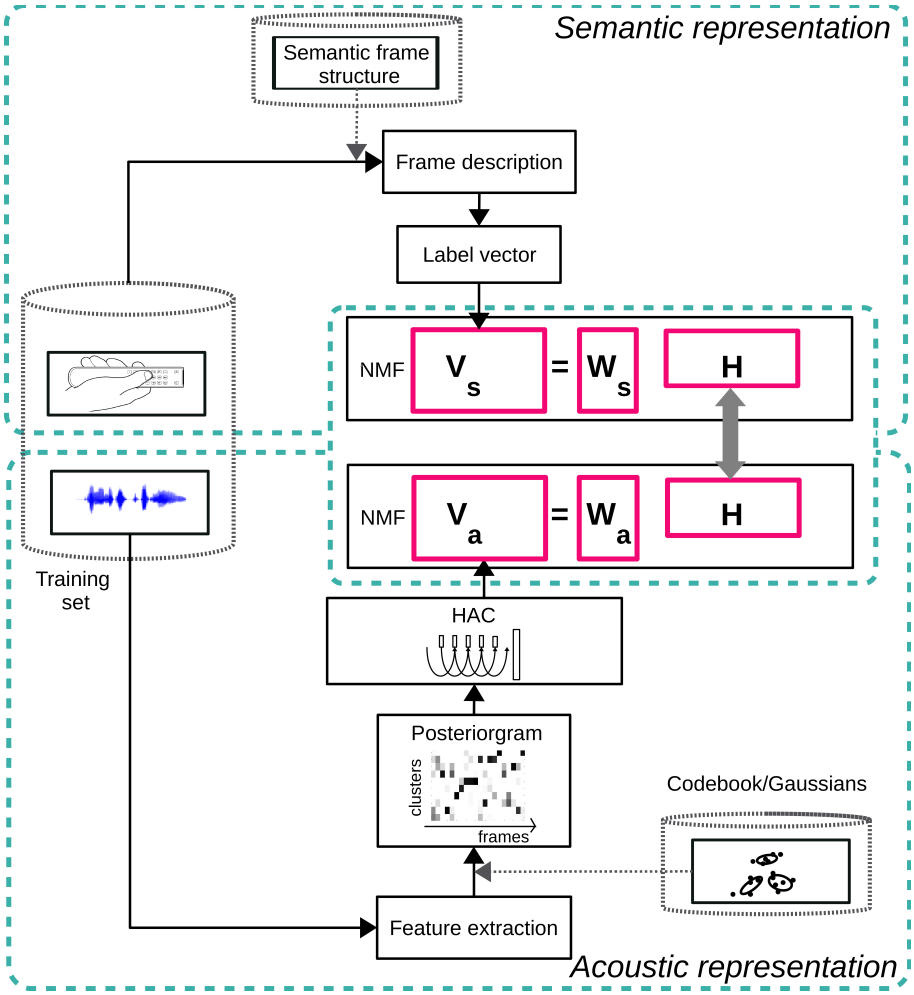


Figure 5.1: *A schematic overview of the learning framework. Two data streams containing acoustic (lower part) and semantic (upper part) features are enhanced and processed in the direction of the arrows towards the centre where they are combined using NMF.*

on recurrent patterns which are co-occurring between the two data-streams. The finding of recurrent patterns, co-occurring between the two data streams, is laying at the heart of the learning procedure (cf. section 5.4.3), where idiosyncratic expressions are parsed and linked to operations on a device. The

steps and algorithms are explained with more detail in the following sections.

5.4.1 Semantic representation

A *semantic frame* [19] is a data structure that represents the semantic concepts in a spoken utterance which users are likely to refer when they control a device by voice. Each semantic frame is composed of slots, which in turn contain slots or values. Different commands with a similar structure are represented by the same semantic frame structure but use different slot values. For example, the correspondence between commands like "Switch off the kitchen light" and "Switch on the bathroom light", could consist of a switching *action* on the *object*, here "light", at a particular *location*. A semantic frame with three slots, labelled by <`action`>, <`object`> and <`location`> is a possible generic structure for parsing such commands. The values in the slots relate to the concepts describing the intended setting. Each slot allows the selection of one value from a predefined list, such as <`on, off, ...`>, <`lights, ...`> and <`kitchen, bathroom, ...`> in this example. The values also relate to the functionality of the devices and this can be understood as a place holder with the potential capacity to hold a spoken word or phrase referring to a relevant concept. For instance, the semantic frame structure in the example above also covers commands like "Turn on the light in the kitchen", where the spoken phrase "Turn on" is related to the value <`on`>. The challenge is in learning to distinguish the semantic frame and filling in the correct values in the relevant slots, allowing the user to choose his own words and using his own pronunciations.

The semantic frame description of the n^{th} utterance is converted into a binary *label vector*, denoted by $\mathbf{v}_{s,n}$, indicating the presence or absence of slot-values collected in all frames and slots. It is a fixed-length column vector with L entries equal to the total number of slot values. Note that multiple slot values are likely to be active in a single utterance and that their presence is highly correlated since the same slot values are usually active in different repetitions of the same command. Sorting all active label entries is a multi-label classification problem as multiple labels are decoded at the same time. For the collection of N utterances in the training set, the semantic representation is composed as $\mathbf{V}_s = [\mathbf{v}_{s,1}\mathbf{v}_{s,2}\dots\mathbf{v}_{s,N}]$. A second utterance-based representation is built from the acoustic features as explained in the following section.

5.4.2 Acoustic representation

Feature extraction

The first steps in the feature extraction method are pre-emphasis and windowing followed by the fast Fourier transform. The obtained physical frequencies are rescaled to mel-frequencies which are believed to emulate the frequency scale of the human ear [20], that is approximately a linear frequency spacing below 1 kHz and a logarithmic spacing above 1 kHz. Mel-spectral magnitudes are logarithmically scaled as well and transformed into cepstral coefficients (MFCC features) by using the (inverse) discrete cosine transform. Other standard procedures in the preprocessing phase consists of voice-activation detection to remove silence frames, and utterance-based mean and variance normalisation.

Codebook training

The acoustic frames are partitioned into clusters by using a codebook training procedure adopted from [17]. The procedure starts with one cluster and iteratively splits the cluster with the lowest frame sample density into subclusters. The clusters and the frames are repartitioned at each split iteration using k-means clustering. The Euclidean distance between frames was used as distance measure. The procedure continues until the requested number of K clusters is obtained. The codebook training procedure is followed by the estimation of a full-covariance Gaussian for each cluster. The set of clusters is denoted by Φ and $j = 1 \dots K$ with K the cardinality of Φ. It is evidenced in [21] that speaker-dependent codebook training on smaller training sets are more effective than codebook training using larger training sets with speech pooled from different speakers. Therefore, we opted to use speaker-dependent codebooks for each speaker in this study.

Posteriorgram

A posteriorgram $\mathbf{P}_{t_i,\theta_j}$ is a two dimensional data structure $(K \times Q)$ containing the posterior probabilities that the observation in the frame at time t_i, with $i = 1 \dots Q$ and Q the number of frames, is drawn from the cluster θ_j under the assumption of a Gaussian distributed cluster membership. The posteriorgram provides a soft localization of the frame with respect to all the cluster locations in the feature space and it contains positive values for which only a few are substantially different from zero.

Histogram of acoustic co-occurrence

The posteriorgram of an utterance has a variable length depending on the number of frames in the utterance while a fixed-length vector is required to compose a data matrix that is suitable for NMF. The aim of HAC [22] is twofold. HAC representations allow building fixed-length vectors for each utterance by accumulating the probability of observing the clusters (θ_a, θ_b) over two frames, shifted τ frames away from each other. The number of clusters is a constant, therefore, all possible $K \times K$ co-occurring combinations for (θ_a, θ_b) is constant too. Secondly, the HAC of a posteriorgram is robust against small temporal variations because the HAC features consist of soft counts of co-occurring frames within small time delays (up to 20 frames in this study). Representations with an absolute time reference like posteriorgrams would be more prone to time-dependent variation urging the use of time warping algorithms to compute the alignment between two time series. For the n^{th} utterance spanning Q frames, the co-occurrence soft count over a time delay τ for the cluster pair (θ_a, θ_b) in $\Phi \times \Phi$, is defined as follows (see [23])

$$[\mathbf{v}_n^\tau]_{(\theta_a,\theta_b)} = \sum_{t_i=0}^{q-\tau} \mathbf{P}_{t_i,\theta_a} \mathbf{P}_{t_i+\tau,\theta_b} \tag{5.1}$$

and $\forall t_i, i = 1 \ldots Q, \sum_{\theta \in \Phi} \mathbf{P}_{t_i,\theta} = 1$.

The HAC is an accumulation of all the Gaussian co-occurrence probabilities denoted by the row vector $\mathbf{v}_n^\tau$. The information captured by the HAC depends largely on the chosen delay by which two frames are separated from each other in time. Therefore, multiple time aspects are incorporated by stacking HAC's with shorter and longer delays to reach both within and across words and word boundaries. As a result, a large fixed-length column vector is built, denoted as $\mathbf{v}_{a,n} = \left[\begin{array}{cccc} \mathbf{v}_n^{\tau_1} \mathbf{v}_n^{\tau_2} \ldots \mathbf{v}_n^{\tau_C} \end{array} \right]^T$, where C represents the number of HAC's. Analogously to the semantic denotation $\mathbf{V}_s$, the acoustic representation is composed as $\mathbf{V}_a = [\mathbf{v}_{a,1} \mathbf{v}_{a,2} \ldots \mathbf{v}_{a,N}]$ for the collection of N utterances in the training set.

5.4.3 Non-negative matrix factorization

NMF decomposes a data matrix into the product of two low-rank matrices; one factor $\mathbf{W}$ represents latent structure, that is recurring patterns in the columns of $\mathbf{V}$, the second factor $\mathbf{H}$ indicates which columns in $\mathbf{W}$ (patterns) are combined to approximate the columns in $\mathbf{V}$. In simultaneous NMF ([24]), data from different modalities are factorized simultaneously, leading to recurrent

patterns in the columns of $\mathbf{W}$ consisting of pattern combinations over two or more sources that coincide with each other. Many names have been used for the same multimodal factorization algorithm depending on the kind of source material, like for instance joint NMF in [25], or when one stream consists of supervision data, it has been referred to as semi-supervised NMF [26] or weakly supervised NMF [22].

Here, we jointly factorize the semantic and the acoustic representation in order to find the acoustic patterns that co-occur with the active label entries. The joint factorization of $\mathbf{V}_s$ (cf. section "5.4.1") and $\mathbf{V}_a$ (cf. section "5.4.2") is expressed as follows:

$$\begin{bmatrix} \mathbf{V}_s \\ \\ \mathbf{V}_a \end{bmatrix} \approx \begin{bmatrix} \mathbf{W}_s \\ \\ \mathbf{W}_a \end{bmatrix} \mathbf{H} \tag{5.2}$$

where $\mathbf{W} = [\mathbf{W}_s \mathbf{W}_a]^T$ and $\mathbf{H}$ are two matrices of lower rank. The co-occurring semantic and acoustic patterns are found in $\mathbf{W}_s$ and $\mathbf{W}_a$ respectively. The n^{th} column in $\mathbf{H}$ describes which co-occurring patterns are active in the n^{th} utterance. The inner dimension in the right half of Eq. 5.2 determines the number of co-occurring patterns in which the dataset is decomposed. It is usually a low number in a small vocabulary task since it reflects the number of slot values L in the VUI. However, by increasing the inner dimension with a number D, columns are introduced in $\mathbf{W}$ to represent patterns for filler words, i.e. recurrent acoustic patterns such as "please" or "the" that are usually left out in the semantic representation.

The latent patterns are found by minimizing the difference between both sides of Eq. 5.2. Since $\mathbf{V}_s$ and $\mathbf{V}_a$ consist of (soft) count data, the Kullback-Leibler divergence [27] is preferred as loss function and is expressed as follows:

$$(\mathbf{H}^*, \mathbf{W}_a^*, \mathbf{W}_s^*) = \underset{(\mathbf{H}, \mathbf{W}_a, \mathbf{W}_s)}{\arg\min} D_{KL}\left(\begin{bmatrix} \mathbf{V}_s \\ \\ \mathbf{V}_a \end{bmatrix} || \begin{bmatrix} \mathbf{W}_s \\ \\ \mathbf{W}_a \end{bmatrix} \mathbf{H} \right) \tag{5.3}$$

Iterative update rules for minimizing a distance measure between the left- and the righthand side can be found in [27]. It has been demonstrated that convergence is guaranteed towards a local optimum. Note that the loss function in Eq. 5.3 can be seen as a regularization in which acoustic patterns are preferred that correspond to the occurrences of slot values. Writing the loss function in Eq. 5.3 as a regularised loss function results in:

$$(\mathbf{H}^*, \mathbf{W}_a^*, \mathbf{W}_s^*) = \underset{(\mathbf{H}, \mathbf{W}_a, \mathbf{W}_s)}{\arg\min} \left[D_{KL}\left(\mathbf{V}_a || \mathbf{W}_a \mathbf{H}_a\right)) + \lambda D_{KL}\left(\mathbf{V}_s || \mathbf{W}_s \mathbf{H}_s\right) \right] \tag{5.4}$$

with $\lambda = 1$ and $\mathbf{H}_a = \mathbf{H}_s$ for equivalence with Eq. 5.3. If $\mathbf{H}_a$ and $\mathbf{H}_s$ are allowed to be tied loosely, then an additional regularisation term should be added to minimize the difference between $\mathbf{H}_a$ and $\mathbf{H}_s$, which was pursued in [25].

5.4.4 Recognition

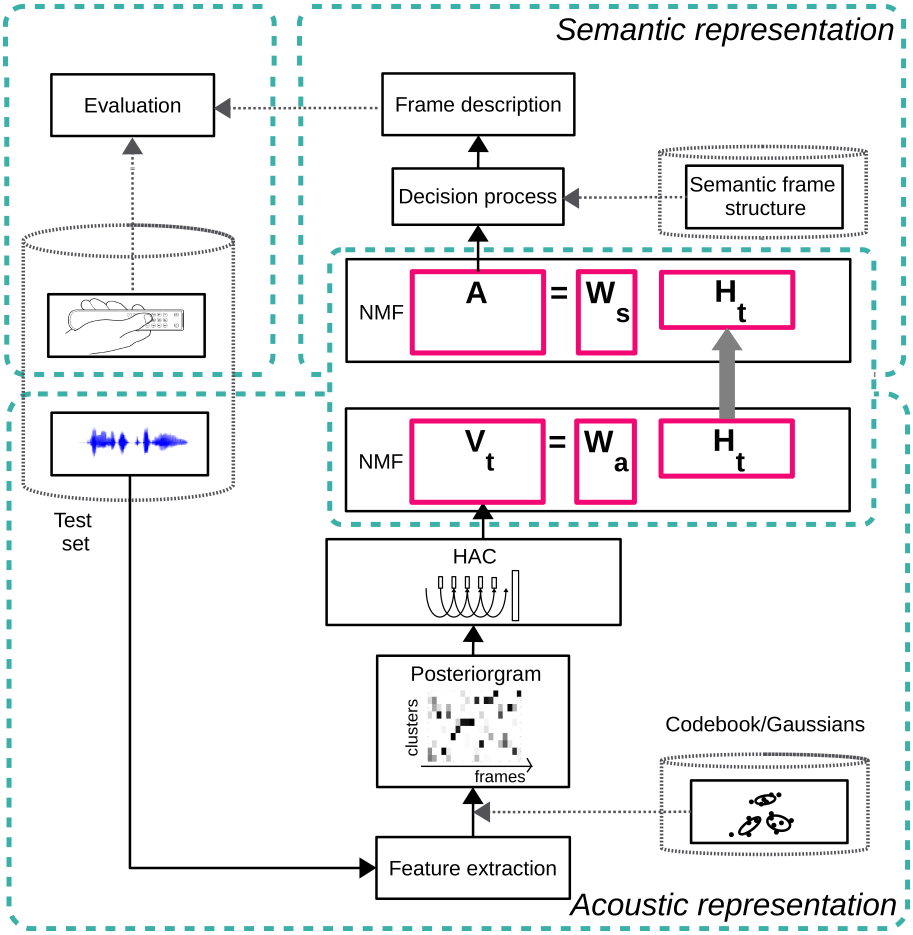


Figure 5.2: *A schematic overview of decoding. Only acoustic data is available and the processing proceeds from the bottom to the upper part where a decision process takes place to validate the interdependent activations of different words.*

The aim of the VUI is to find the frame description for a spoken utterance. A schematic overview of the recognition phase is depicted in Figure 5.2. Speech processing of a command proceeds from the spectro-temporal representation in the lower part of Figure 5.2 to the HAC representation in the centre, after which NMF takes place in order to obtain the load matrix $\mathbf{H}_t$ using the learned

patterns in $\mathbf{W}_a$ that were co-occurring with the semantic patterns in $\mathbf{W}_s$ in the training phase. $\mathbf{H}_t$ is then transferred to the upper part of Figure 5.2. The slot value activations $\mathbf{A}$ are found by using $\mathbf{H}_t$ and using $\mathbf{W}_s$ obtained in the learning phase. Finally, the arrow leaving from the box: "Semantic frame structure" indicates that the semantic structure is superimposed on slot value activations as a decision process where groups of slot values are compared and related to each other (cf. section 5.4.4). Knowing the correct frame description of the spoken command allows for the proper execution of the command.

Activation

We denote the data matrix and the load matrix in the test phase by $\mathbf{V}_t$ and $\mathbf{H}_t$. The data matrix $\mathbf{V}_t$ contains the processed speech signal and $\mathbf{H}_t$ is found by minimizing the Kullback-Leibler divergence between $\mathbf{V}_t$ and the matrix product of the acquired $\mathbf{W}_a^*$ and the unknown $\mathbf{H}_t$.

$$\mathbf{H}_t^* = \arg\min_{\mathbf{H}_t} D_{KL}(\mathbf{V}_t||\mathbf{W}_a^*\mathbf{H}_t) \tag{5.5}$$

Contrary to Eq. 5.3, an optimal solution for $\mathbf{H}_t^*$, given $\mathbf{W}_a^*$, is guaranteed since the loss function in Eq. 5.5 expresses a convex problem. The obtained matrix $\mathbf{H}_t^*$ and the acquired matrix $\mathbf{W}_s^*$ are used to provide the slot value activations in $\mathbf{A}$,

$$\mathbf{A} = \mathbf{W}_s^*\mathbf{H}_t^* \tag{5.6}$$

Note that the last step in Eq. 5.6 allows the freedom to obtain slot value activation from different latent factors in $\mathbf{W}^* = [\mathbf{W}_s^*\mathbf{W}_a^*]^T$. A slot value activation can depend on one latent factor or a combination of latent factors in $\mathbf{W}^*$.

Decision process

The decision whether a particular slot value applies or not, depends on the ensemble of activations signalling the presence of the related frame, slots and values. The interdependent relation between frames, slots and values is vital information that we use on top of the activations obtained from Eq. 5.6. The whole semantic frame structure is taken into account by implementing a tree structure for each frame and a few activation spreading rules. In the tree structure (see 5.3), the frames are considered root nodes, slots compose branch nodes and slot values compose leaf nodes. Activations in leaf nodes

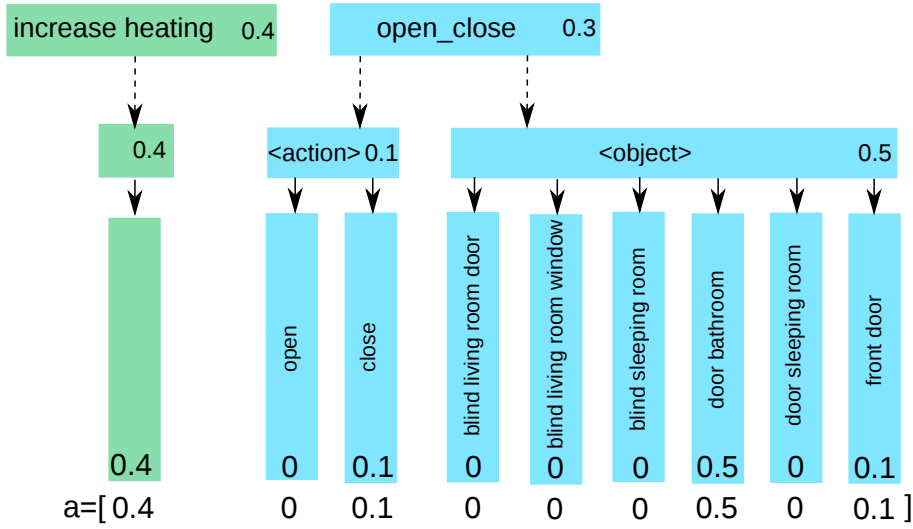


Figure 5.3: *A parse tree of the first two frame descriptions listed in Table 5.6 and the propagation of activation depending on exclusive and selective relations. See text for more explanations.*

correspond with the activations in the columns of $\mathbf{A}$, denoted by the vector $\mathbf{a}$ and representing the NMF activations of one utterance. For each entry in $\mathbf{a}$, there is one leaf node*. The activation spreading rules enable values in $\mathbf{a}$ to propagate to slot and frame levels. These activation spreading rules bear on child-parent relations to which we refer as "exclusive" and "selective" relations. In an *exclusive* relation, only a predefined number of entries, denoted by the constant U, occur at the same time. For instance, the light can be switched on or off, but not both at the same time, therefore, a command can only be assigned one truth value ($U = 1$) in the list <`on, off`>. Generally, if we denote the activation of a parent node by a_p and the activations of its child node by a_{ci}, with $i = 1, \ldots, z$ and $a_{c1} > a_{c2} > \ldots > a_{cz}$, then the following activation spreading rule applies for an exclusive relation: $a_p = median\left(\{a_{ci} | a_{ci} \geq a_{cU}\}\right)$. A selective relation differs from an exclusive one in the precognition of the number of valid child nodes. If the number of valid child nodes is unknown, then activations are compared against a threshold. The following general activation spreading rule applies for a selective relation: $a_p = median\left(\{a_{ci} | a_{ci} \geq a_0\}\right)$ and a_0 is a threshold determined by the p^{th} percentile of all activations in $\mathbf{a}$.

*Note that frames without slots should have a corresponding entry in $\mathbf{a}$, which determines the activation scores of the empty slot value

Multiple frame descriptions are in competition and the frame description with the highest activation in the root node is selected. The U highest activated slots and values for exclusive relations and the slots and values scoring higher than a_0 for selective relations are included in the predicted frame.

A toy example is depicted in Figure 5.3 where the first nine entries of $\mathbf{a}$ are set to $[0.4, 0, 0.1, 0, 0, 0, 0.5, 0, 0.1]^T$ corresponding to the frame description listed in the upper half of Table 5.6. Exclusive relations are depicted as solid arrows and selective relations are depicted as dashed arrows. The first slot value has activation 0.4 and corresponds to the empty slot value of the frame "increase heating". The activation is propagated to the slot level and from there to the frame level. Exclusive relations are presumed for the slots <`action`> and <`object`> and their respective slot values. A preset value of $u = 1$ for both slots yields the propagation of the highest activation, $a_p = 0.1$ and $a_p = 0.5$, to the <`action`> and <`object`> slots, respectively. The relation between the <`open, close`> frame and its slots is evaluated as selective, and by assuming a preset threshold beneath 0.1, their median, $a_p = 0.3$, is propagated to the frame level <`open, close`>. Generally, the median is an unbiased measure for propagating activations when the number of slots and values differs between different frames. In this hypothetical example, the frame "increase heating" and "open_close" have activation 0.4 and 0.3 respectively, thus, the frame "increase heating" and its selected slots and slot values are the predicted outcome of the spoken utterance yielding the activations $\mathbf{a}$ in this toy example.

5.5 Reference model

We compared NMF learning with Gaussian mixture models (GMM). It is hard to preset the number of GMM components especially when data sets are small and have varying sizes among speakers (see subsequent section 5.6). Therefore, we investigated four GMMs having $10, 20, 40$ or 80 components fixed for each speaker, respectively, with a diagonal covariance structure instead of a full one in order to limit the number of free parameters. The GMMs were embedded in the architecture of our framework in a similar way as the NMF learning module. At the front end, feature extraction was identical up to and including the posteriorgram step (see Figure 5.1), after which a scaling step was introduced using the logit function, $\log p/(1-p)$, to map the posterior probabilities in the posteriorgram to the real line $\mathbb{R}$. Subsequently, utterances in the data with a common semantic entry, i.e. utterances with "1" at a particular position in the label vector $\mathbf{v}_s$, were pooled together to compose the training set for GMM estimation of each respective slot value. Thus, for each label entry in $\mathbf{v}_s$ there is one GMM predicting the presence of the respective slot value in the decoding

phase. Similarly to the NMF activations (see section 5.4.4), the posterior probabilities are committed to the same decision process (cf. section 5.4.4). It should be noted that GMMs do not capture temporal dependencies while HAC's do. A GMM can be conceived as a Hidden Markov Model (HMM) with one state per slot value. By using a HMM with multiple states and tuning transition probabilities, temporal relations among the acoustic features can be captured. However, it is evidenced in [28] that GMMs outperform HMM's in accuracy for small training sets.

5.6 Speech material

Similar to [18], two datasets are employed. The first dataset is PATCOR containing recordings of 10 speakers playing a solitaire card game by voice. The second dataset is a recent recorded dataset called DOMOTICA-3 dubbing its precursor DOMOTICA-2 employed in [18]. The utterances consist of commands controlling a home automation system by voice.

PATCOR

Table 5.1: Participants in PATCOR

Pid	*gender*	*Age (years)*	*Wizard-of-Oz*	*number of games*	*number of utterances*
1	♀	33	yes	6	274
-	♀	41	yes	2	169
2	♂	45	yes	4	260
3	♂	42	yes	5	278
4	♀	23	no	4	222
5	♀	26	no	4	248
6	♂	24	no	4	223
7	♂	26	no	4	240
8	♀	73	no	5	235
9	♀	22	yes	5	262

The database PATCOR contains recordings of subjects playing the card game "patience" on computer, using only spoken commands. The database contains 10 speakers with more than two thousand commands. The data was collected from unimpaired subjects with non-pathological speech, speaking Belgian Dutch. As depicted in Table 5.1, six participants were females and the age ranged between 22 and 45 years old for almost all speakers except for speaker 9 who was 73 years old. All players played at least four games leading to 254 recorded utterances on average, except for the speaker in the second entry who played only two games.

In order to provoke commands in a natural human-machine like interaction, a wizard-of-Oz setup was employed for five players as indicated in column four of Table 5.1. In a wizard-of-Oz setup, a subject is deceived to believe that the machine is able to commit responsive behaviour, while, in reality, the administrator is taking care of the responsive actions of the machine. The five other players in PATCOR were committed to the same procedure, however, they were told that the administrator took care of all the actions.

The users were free to choose their own words and grammars allowing different expressions for the same card move. A typical utterance in PATCOR is *"Put the four of clubs on the five of hearts"*. The standard frame structure of the utterances used in [18] are demonstrated in Table 5.4.

Domotica-3

Table 5.2: Synoptic description of all actions in Domotica-3, partitioned in columns according to frame type.

increase_heating	***open_close***	***ranged***	***on_off***
increase_heating	close_blind_living_room_door	dimstate1_floor_lamp	off_light_living_room
	close_blind_living_room_window	dimstate2_floor_lamp	off_light_sleeping_room
	close_blind_sleeping_room	dimstate3_floor_lamp	off_lights
	close_door_bathroom	level1_head_bed	on_light_bathroom
	close_door_sleeping_room	level2_head_bed	on_light_kitchen
	close_front_door	level3_head_bed	on_light_living_room
	open_blind_living_room_door		on_light_sleeping_room
	open_blind_living_room_window		on_reading_light
	open_blind_sleeping_room		
	open_door_bathroom		
	open_door_sleeping_room		
	open_front_door		

The DOMOTICA-3 database contains Dutch, dysarthric speech commands related to home automation. A typical DOMOTICA-3 utterance is "*turn on the kitchen light*". The dataset contains recordings of the speakers that participated in the collection of the DOMOTICA-2 dataset in [18].

In short, a two-phase data collection method was used in DOMOTICA-2. In the first phase, nine users were asked to command 29 distinct actions in a 3D home environment on computer, guided by a visualised and narrative scenario such as *"you enter the kitchen, but it is dark... "*, in order to provoke an action, but to ensure an unbiased choice of words and grammar. Consequently, each user produced a list of natural induced commands, thus, nine different lists of commands controlling the same actions were created. Some participants missed out a few actions during the guidance of the narrative scenario, but never more than two. The lists, all counting 27 to 29 commands, were read

repeatedly by multiple speakers who participated in the DOMOTICA-2 data collection. A selection of 27 actions from the DOMOTICA-2 collection (see Table

Table 5.3: Participants in Domotica-3

Pid	*gender*	*age (years)*	*profile*	*spoken list number*	*number of utterances*	*Intelligibility score*
17	♀	25	Spastic Quadriparesis	6	347	88.6
28	♀	42	Severe nasal dysarthria	6	204	73.1
29	♂	44	Spastic Quadriparesis	7	174	73.6
30	♂	33	Ataxic dysarthria	5	198	69
31	♂	11		8	225	
32	♀	43	Mild dysarthria and hyperkinetic speech	4	41	65.6
33	♂	33	Ataxic dysarthria, short phonation	3	113	66.2
34	♂	61	Multiple sclerosis	2	331	76.2
35	♀	25	Spastic Quadriparesis	6	268	72.3
37	♂	10		8	156	
40	♂	55	Myotonic-flaccid dysarthria	1	184	85.5
41	♀	39	Dysarthria	2	144	64.2
43	♀		Multiple sclerosis	1	133	89.4
44	♂		Multiple sclerosis	9	164	89.2
46	♀	50	Multiple sclerosis	1	97	74.9
47	♂		Multiple sclerosis	7	64	73.4
48	♂		Multiple sclerosis	5	169	85.8

5.3) were used in the new recordings of the DOMOTICA-3 database. A recording session lasted more or less an half hour in which the speaker read repeatedly the commands from one of the nine lists (see fifth column in Table 5.3). To keep correspondence with previous and future work, we refer to these speakers by unique id's. For all adult speakers, speech intelligibility scores were obtained by analysing the recorded speech using the automated procedure in [29]. While a score above 85 is considered as normal speech intelligibility, a score equal to or below 70 is considered as severely impaired. Speaker characteristics are listed in Table 5.3. Speaker 31 and 37 were children and did not conduct an intelligibility test. Additionally, speaker $43, 44, 46, 47$ and 48 were diagnosed as multiple sclerosis patients and some of them demonstrated adequate speech intelligibility. They were recruited because the digressive nature in time of their speech ability would allow for speech-degenerating data collection in the future. Most speakers were able to generate six or more repetitions of the command lists, except for speaker 31 and 47 who were able to produce one and two repetitions, respectively. A few speakers received a reduced list with 10 commands with at least 10 repetitions. A larger number of repetitions allow us to investigate whether learning improvements proceed even further by adding more learning examples, or whether it levels off at a particular stage. The number of utterances are indicated in column six of Table 5.3. The frame description used in [18] are displayed in Table 5.6.

The database contains speech recorded in realistic environments with two microphones. One microphone was a head-worn set C520 and the other one was a R*O*DE M2 live condenser microphone, which was located in front of the speaker on top of a table at about 50 to 100 cm. The recordings were held in a room selected in the respective health care centre of the patient which range from quiet to some background speech. The recordings were carried out with a sampling rate of 48 khz and a resolution of 24 bit for each channel after which it was degraded to a sampling rate of 16 khz and stored as such in the corpus. The recordings of speaker 33 and 40 barely reached voice activation levels because the directed microphone of the headset was too far out of reach, however, the recordings on the second channel did succeed.

5.7 Hierarchical knowledge representation

Frame *(exclusive)*	***Slot*** *(selective)*	***Value*** *(exclusive)*
dealcard	-	-
movecard	<from_suit>	c,d,h,s
	<from_value>	1-13
	<from_foundation>	1-4
	<from_column>	1-7
	<from_hand>	-
	<target_suit>	c,d,h,s
	<target_value>	1-13
	<target_foundation>	1-4
	<target_column>	1-7

Table 5.4: PATCOR - compositional. Here, the letters c,d,h and s represent the suits clubs, diamonds, hearts and spades, respectively.

An optimal structure depends on different factors like the number of decision steps in the recognition process, thus the number of levels in the hierarchy (cf. section 5.4.4). It also depends on the number of alternatives at each step. These factors are not independent from each other. For instance an ordered tree with more levels will induce more decisions, but with lower complexity. The kind of decision rule also plays a role of importance. In this study, we explore the influence of the semantic frame composition on the recognition performance by considering two different frame structures employing different hierarchical levels and decision rules.

Frame *(exclusive)*	***Slot*** *(selective)*	***Slot*** *(exclusive)*	***Slot*** *(selective)*	***Value*** *(exclusive)*
dealcard	-	-	-	-
movecard	<from>	<card>	<suit>	c,d,h,s
			<value>	1-13
		<foundation>	-	1-4
		<column>	-	1-7
		<hand>	-	-
	<target>	<card>	<suit>	c,d,h,s
			<value>	1-13
		<foundation>	-	1-4
		<column>	-	1-7

Table 5.5: PATCOR - hierarchical. the letters c,d,h and s represent the suits clubs, diamonds, hearts and spades, respectively.

Frame *(exclusive)*	***Slot*** *(selective)*	***Value*** *(exclusive)*
increase heating	-	-
open_close	<action>	open,close
	<object>	1-6
ranged	<range>	1,2,3
	<object>	1,2
on_off	<action>	on,off
	<object>	1-6

Table 5.6: DOMOTICA-3 - compositional. The lower panel pertains to the DOMOTICA-3 database where the numbers 1-6 refer to objects such as a kitchen lamp or a bathroom door.

We will investigate two approaches for the validation of frame structure on the database PATCOR; one is the compositional standard shown in Table 5.4 and employed in [18]; the second one is a hierarchical semantic frame structure with one additional level shown in Table 5.5. The decision rules for each layer are listed in the column headings in italic font style. In the standard description, commands are decomposed into parts such as suits, values and columns. In the *compositional* structure, a selective rule is used to compare the activations of alternative slots against a threshold. Known information is left unexplored like for example the impossible co-occurrence of a card moved from the hand and from the foundation. A second structure is called *hierarchical* referring to more levels in which slots contain slots or values. Such a structure eases the decision

step as the number of alternatives is limited in each layer with no more than two alternatives in selective decisions.

For the DOMOTICA-3 dataset, we employ even more distinctive structures on the semantic representation. The first structure entails the mapping of entire spoken commands to frames without slots or values, leading to a scenario where the machine learning problem reduces to a multi-class paradigm, that is one class for each possible command. Clearly, such a mapping is unattainable for sets with complex commands as in PATCOR, but, for a small set of commands, modelling entire utterances is a viable option. Note that such a structure is less robust to word order variation and alternative expressions of the same command when utterances are learned in their entirety. We compare a semantic frame structure with commands modelled in their entirety with a compositional approach which parses commands into parts such as objects and actions [18]. This semantic frame structure is shown in Table 5.6. The values 1 to 6 refer to objects or devices such as kitchen lamp or bathroom door. Once again, we expect improved performance for the multi-layered frame structure since selective rules are used for layers holding only two slots, while multiple alternatives are gathered in levels with exclusive rules.

5.8 Experiments

The goal of the experiments is twofold: first, we test the feasibility of our VUI by evaluating the performance of the framework using the F-score on slot value recognition as defined in [18], furthermore, we investigate the added value of using a more layered semantic frame structure on two datasets; PATCOR containing commands having a complex grammar and DOMOTICA-3 containing realistic recordings of commands from speech-impaired speakers in the setting of a virtual home automation system.

An important feasibility issue is the speed of learning, which we evaluate by tracking the gain in slot value recognition for incrementally increasing training sets. This procedure allows us to plot a learning curve, that is, the curve representing the average slot value recognition score in function of the average number of learning examples. The rate of learning is usually sharpest in the beginning and gradually evens out against an asymptotic level. We are especially interested in the initial and final phase of the learning curve; on the one hand, the speed of learning should be high so users gain interest in keeping on using the VUI, thus keeping on training the system; on the other hand, the learning curve should not level off to low, that is, the VUI should not get stuck in suboptimal functionality in the long run. Clearly, the speed of learning and the asymptotic performance are important attributes of a useful learning procedure.

5.8.1 Setup

Evaluation procedure

The data was partitioned in blocks containing approximately an equal number of slot values using an algorithm outlined in [18]. This algorithm minimises the Jensen-Shannon divergence between the slot value distributions over all blocks. Likewise [18], block creation was followed by the composition of a Latin square from which the first five rows were submitted to a five-fold cross-validation experiment. In each fold or row of the Latin square, the first x blocks were used as train set while the remaining $Z - x$ blocks were used as test set with Z the total number of blocks. While the train sets increased incrementally with one block, $x = 1, ..., Z - 1$, the test sets decreased decrementally with one block. The incrementally increasing training sets allowed us to evaluate the learning performance at different time stamps in the learning process of the VUI. The slot values that appeared in each block at least once were used for scoring in the test sets. Note that the real performance of the vocal interface also depends on the interface's ability to distinguish between commands and other utterances spoken in a domestic environment. Here, we focus on the rate of learning assuming a perfect classification of commands directed to the system against utterances that were not.

For the evaluation of the framework, we excluded the speaker without Pid number in Table 5.1 in PATCOR and speaker 32 and 47 in DOMOTICA-3, due to data insufficiency for block creation. In addition, we created two groups for the DOMOTICA-3 corpus in order to evaluate the feasibility of the framework. Speakers $29, 30, 33, 41$ and 46 have an intelligibility score below 75 and uttered less than 200 commands. We refer to this group as *severe dysarthria*. Note that an intelligibility score higher than 85 is not considered pathologic. Speakers $17, 28, 31, 34$ and 35 were joined in another group because they uttered more than 200 commands allowing us to track the performance of the system in the long run.

Parameters

We used pre-emphasis ($\alpha = 0.97$, sampling rate at 16 khz) and Hamming windowing with 30 ms frames in addition to a frame shift of 10 ms. 14 cepstral dimensions were retained and the first and second order differences were appended leading to 42 feature dimensions. Silence frames were removed before the codebook training started, aiming for $K = 100$ clusters from which posteriorgrams were obtained with 100 entries. The main portion of the probability mass in a frame seems to originate from only a few clusters, therefore,

we retained only the three highest probabilities in each frame in order to gain computational efficiency by using sparse matrices of HAC features. We stacked $C = 4$ HAC's with delays $\tau = 2, 5, 9$ and 20 resulting in $4 \times 100^2 = 40000$ entries for each utterance-based acoustic representation $\mathbf{v}_a$.

$\mathbf{H}_{init}$ and $\mathbf{W}_{init}$ denote the initialisation of $\mathbf{H}$ and $\mathbf{W}$ respectively

$$\mathbf{H}_{init} = \begin{bmatrix} \mathbf{V}_s + \lambda \mathbf{A}(R \times N) \\ \\ \mathbf{B}(D \times N) + \gamma \mathbf{1}(D \times N) \end{bmatrix} \tag{5.7}$$

$$\mathbf{W}_{init} = \begin{bmatrix} \mathbf{I}(R \times R) + \lambda \mathbf{O}(R \times R) & \mathbf{P}(R \times D) + \theta \mathbf{1}(R \times D) \\ \\ \mathbf{Q}(F \times (D + R))) \end{bmatrix} \tag{5.8}$$

with D the largest integer smaller than $0.2 \times R$, hence, by way of example, for $R = 40$ slotvalues, $D = 8$ extra columns were added to $\mathbf{W}$. This proportion was constant for all experiments. The parameters λ, γ and θ were set to $1e^{-4}, 0.1$ and 0.2, respectively. All entries in $\mathbf{A}, \mathbf{B}, \mathbf{O}, \mathbf{P}$ and $\mathbf{Q}$ are i.i.d samples from the uniform distribution $\mathcal{U}(0, 1)$ with boundaries $(0, 1)$. $\mathbf{I}$ is the identity matrix and $\mathbf{1}$ is a vector with all ones. The columns of $\mathbf{W}$ were normalised to sum to one throughout the multiplicative updates to prevent drift towards large numbers reducing the cost function.

5.8.2 Results and discussion

Feasibility

Results on DOMOTICA-3 are shown in 5.4 as a function of the number of learning examples in the training set. The depicted results concern recordings on the field microphone. The F-scores for the more severe dysarthric group are depicted in the upper panel. The plotted numbers are the id's of the speakers (see Table 5.3) with circle-shaped lighter and darker gray background colours indicating the NMF-based approach and the 80-component GMM approach, respectively. When we compare GMM-based learning with NMF-based learning in the upper panel, we observe steeper learning curves for NMF-based learning for the group of severely dysarthric speakers yielding an average improvement of 23% ($t_{(159)} = 30.2, p < 0.001$). Moreover, a similar trend can be observed in the group with more training material depicted in the lower panel of 5.4, yielding an average improvement of 20.2% ($t_{(159)} = 38.5, p < 0.001$) by using the NMF-based approach.

We used the non-parametric method in [30, 31] to estimate a smooth learning curve for each speaker using the LOWESS procedure. Optimal smoothness

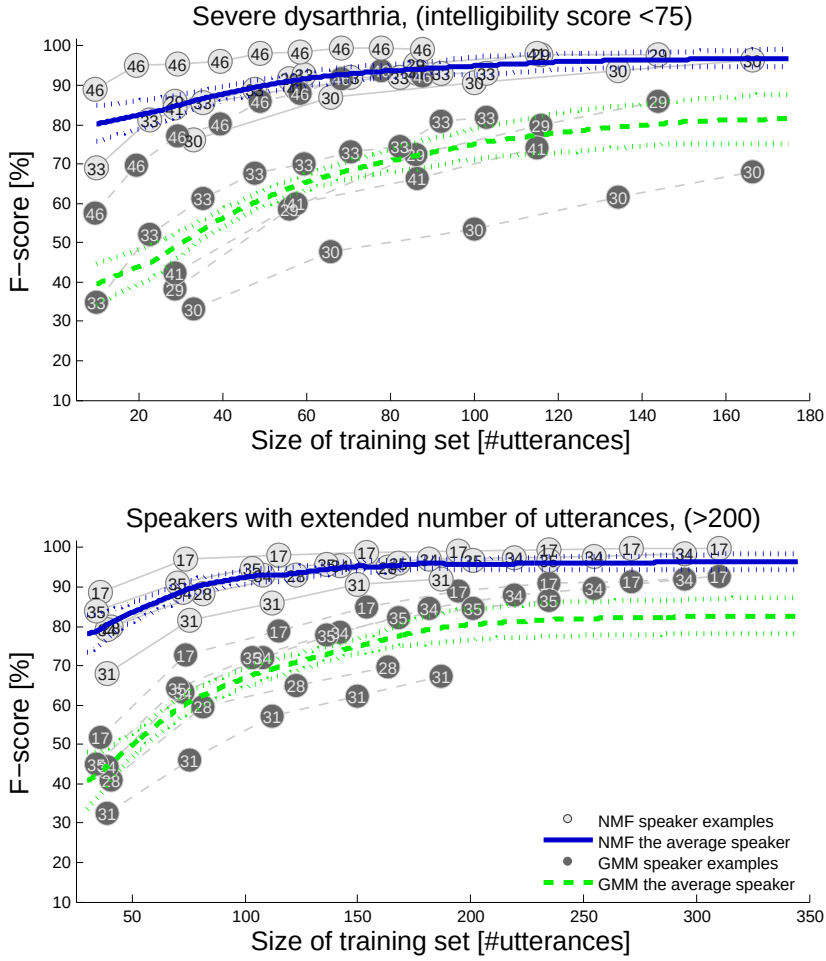


Figure 5.4: *NMF-based learning against GMM-based learning for severe dysarthric speakers in the upper part. Speakers with extended training sets are depicted in the lower part. Numbered circles represent speaker-id and their locations indicate F-scores as a function of the number of utterances in the training sets. Furthermore, the smoothed curves are interpolations of the scattered F-scores using the* LOWESS *procedure and they exemplify the performance of an average speaker.*

parameters were found by cross-validating different smoothness values between 0.4 and 0.8. We plotted the learning curve for the average speaker using a

full blue-coloured and a dashed green-coloured line to indicate the NMF-based and the GMM-based scores, respectively. Furthermore, we constructed 95% confidence limits by bootstrapping the LOWESS procedure and we indicated these bounds by dotted lines. The curves provide an indication how the average speaker is expected to perform.

Table 5.7: *F-scores after* 40 *and* 120 *training utterances for* DOMOTICA-3. *The F-scores are interpolated using the LOWESS procedure.*

				speakers														aver-
				17	28	29	30	31	33	34	35	37	40	41	43	44	46	48	age
DOMOTICA-3 RODE M2	compositional	$N = 40$	GMM 10	60	51	53	41	43	58	52	53	62	43	53	88	45	83	77	57.5
			GMM 80	53	45	47	34	34	65	46	49	56	37	46	87	37	81	79	53.1
			NMF	90	69	83	82	71	87	76	83	84	73	77	99	75	96	98	82.9
		$N = 120$	GMM 10	75	66	66	51	52	67	68	66	69	57	66	88	75	88	85	69.3
			GMM 80	80	65	76	56	59	82	74	73	79	69	68	97	78	96	93	76.3
			NMF	99	88	90	93	86	93	91	94	94	92	96	100	99	97	99	94.1
	flat	$N = 40$	GMM 10	43	37	43	23	25	41	38	43	48	29	35	78	62	73	66	45.6
			GMM 80	24	19	27	18	12	45	23	31	42	16	28	84	57	75	74	38.3
			NMF	88	72	80	76	63	78	79	81	77	71	68	99	98	96	98	81.6
		$N = 120$	GMM 10	65	55	65	39	39	48	61	57	51	51	48	91	88	82	81	61.4
			GMM 80	65	50	74	49	35	67	59	61	65	53	60	97	97	85	87	66.9
			NMF	98	83	95	94	78	85	90	89	88	86	93	100	100	100	99	91.9
DOMOTICA-3 headset	compositional	$N = 40$	GMM 10	58	45	54	42	42	40	45	53	60	35	56	87	59	86	82	56.3
			GMM 80	54	42	49	37	33	51	46	49	58	31	50	87	54	82	82	53.7
			NMF	89	80	89	80	69	79	80	85	86	60	88	99	90	96	98	84.5
		$N = 120$	GMM 10	74	56	69	48	52	43	63	69	65	46	69	87	74	90	88	66.2
			GMM 80	79	65	79	57	58	70	74	75	79	55	75	97	89	96	92	76
			NMF	98	92	97	92	86	92	94	96	95	87	98	100	99	99	100	95
	flat	$N = 40$	GMM 10	41	29	39	24	24	25	36	41	45	25	37	82	70	73	75	44.4
			GMM 80	27	18	32	18	13	31	23	30	38	11	30	85	59	75	74	37.6
			NMF	88	79	90	78	63	66	83	87	79	52	84	98	98	98	99	82.8
		$N = 120$	GMM 10	64	48	67	45	38	32	57	55	48	38	61	80	89	80	80	58.8
			GMM 80	70	45	73	52	39	52	61	58	64	39	63	96	97	82	89	65.3
			NMF	98	88	97	94	82	92	94	92	92	74	98	100	100	100	100	93.4

When comparing F-scores, the NMF approach improves learning up to 40% in the beginning of the learning curve, as can be seen from the difference between the full and the dashed-line averaging curve in Figure 5.4. For the group with severe dysarthria, we observe that the NMF-based approach yields a score close to 80% on average after only one repetition. For instance, speaker 33 has an intelligibility score of 66.2 and yields an F-score of 70% after one repetition and 96% after 9 repetitions. Moreover, some speakers yield scores close to 100% after a few repetitions, like for instance speaker 17 depicted in the lower part of Figure 5.4 obtaining a score above 99% after four repetitions only. Note that the results using the headset recordings are similar as can be seen in Table 5.7. These results are very promising, especially for dysarthric speakers, as both the learning rate and the accuracy are already in a range that is usable for a vocal interface. Moreover, all learning curves who didn't reach a ceiling performance at the end are still rising indicating that with more learning examples the accuracies will probably further improve.

Semantic structure

Here, we compare the results for NMF based learning using two different semantic frame structures; the PATCOR database in the upper panel and the DOMOTICA-3 database in the lower panel of Figure 5.5. When comparing F-scores for the Hierarchical and the compositional frame structure in Figure 5.5, we find a small, but significant overall improvement for using a hierarchical frame structure instead of a compositional one, i.e. $t_{(179)} = 12.4, p < 0.001$, with an absolute average improvement in F-score of 3.3%. The improvements are fairly consistent among speakers despite the fact that the individual scores for the PATCOR database are wide-ranged. The scores are wide-ranged because speakers $3, 5, 7$ and 8 frequently used the words "red" and "black" instead of "hearts", "spades", "clubs" and "diamonds". While the use of colours such as "red" and "black" allows the VUI to distinguish "clubs" and "spades" from "hearts" and "diamonds", it will not allow to learn the difference between the two black or the two red card suits. Since 40% to 50% of the words in the move commands consisted of words referring to the card suits, a drop in overall F-score is observed because the incorrectly recognised card suits are counted as false positives despite the fact that the user did not provide this information in the VUI training. As can be seen in the upper part of Fig. 5, there is a considerable gap between the learning curves of speaker $3, 5, 7$ and 8 using the words "red" and "black" and speaker $2, 4, 6$ and 9 who all preferred the consistent use of the words "clubs", "spades", "hearts", and "diamonds". Another reason for the wide-ranged performances is that some users tend to use a lot of synonyms, which we did not anticipate in the NMF-based approach here. More results on the PATCOR database, including results on GMMs, are listed in Table 5.8.

Note that GMMs with more components perform better if there is enough data to adequately fit all free parameters as can be seen in Table 5.7 and 5.8 when comparing GMM scores for small datasets ($N = 40$) against large datasets ($N = 120$ or $N = 175$). The GMM with 80 components perform better if the dataset set size is equal to $N = 175$. These tables only include GMM scores for 10 and 80 components. The GMM results for GMMs with 20 and 40 components are not reported here because these scores are similar to the 80-component GMM scores.

The corresponding results for the DOMOTICA-3 database are depicted in the lower panel of 5.5 displaying a positive, though, non-significant statistical tendency in favour of the compositional frame structure, i. e. the more profound structure compared to the flat one. The average speaker plot represents scores of all 15 speakers in the database, though, the varying range of results are exemplified by three speakers only for reasons of visibility. A considerable number of speakers yield high F-scores in the beginning while other speakers yield lower F-scores in

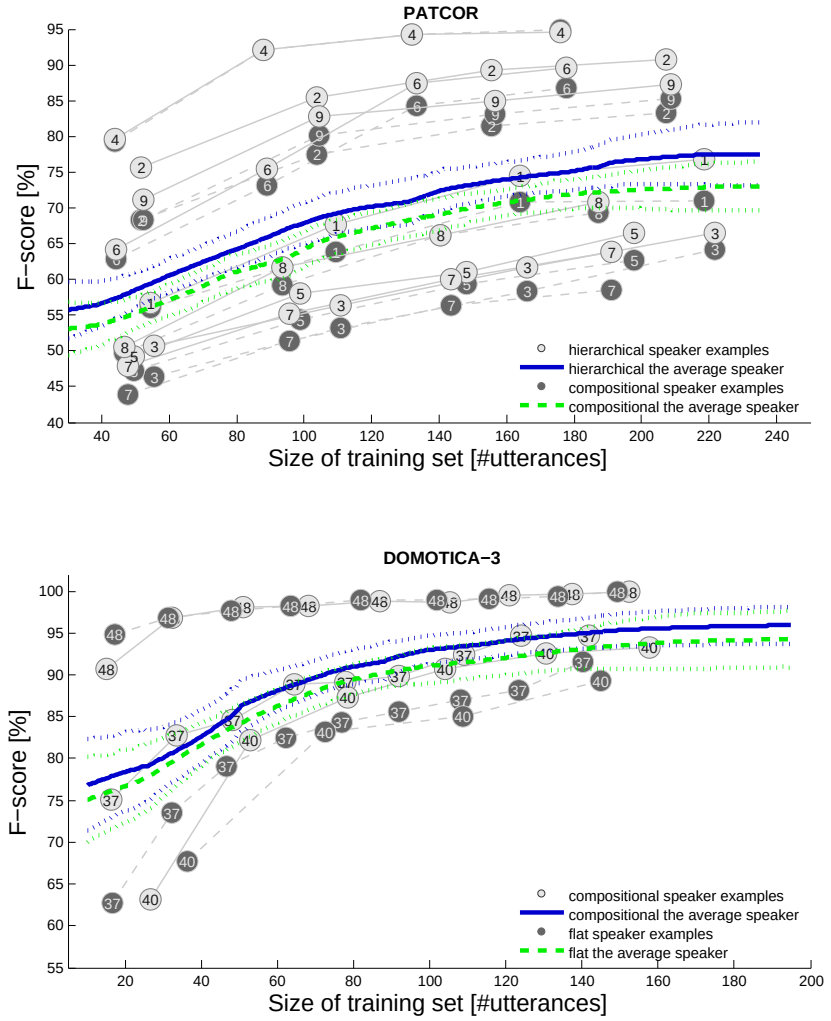


Figure 5.5: *Hierarchical against compositional frame structure for* PATCOR *in the upper part, and the compositional against the flat structure for* DOMOTICA-3 *in the lower part. Numbered circles represent speaker-id and their locations indicate F-scores as a function of the number of utterances in the training sets. Furthermore, the smoothed curves are interpolations of the scattered F-scores using the* LOWESS *procedure and they exemplify the performance of an average speaker*

Table 5.8: *F-scores after* 40 *and* 175 *training utterances for* PATCOR. *The F-scores are interpolated using the* LOWESS *procedure.*

				speakers									aver-
				1	2	3	4	5	6	7	8	9	age
PATCOR	hierar-chical	$N = 40$	GMM 10	38	46	35	49	36	45	34	36	51	41.1
			GMM 80	40	43	38	47	37	43	33	38	46	40.6
			NMF	55	73	50	79	47	64	47	49	69	59.2
		$N = 175$	GMM 10	38	46	35	49	36	45	34	36	51	41.1
			GMM 80	54	71	44	70	46	58	40	43	62	54.2
			NMF	78	91	63	95	63	90	62	68	87	77.4
	compo-sitional	$N = 40$	GMM 10	41	47	35	49	36	45	34	36	50	41.4
			GMM 80	39	43	38	48	37	42	32	38	46	40.3
			NMF	53	66	45	79	46	63	42	49	66	56.6
		$N = 175$	GMM 10	41	47	35	49	36	45	34	36	50	41.4
			GMM 80	54	66	44	70	45	62	40	47	61	54.3
			NMF	72	83	61	95	62	87	58	69	85	74.7

the beginning, but a steeper rise towards the end, as demonstrated by speaker 48 and 37 respectively. The non-significant statistical tendency is probably caused by the ceiling effect, in which a considerable number of speakers have maximum scores for both conditions, making discrimination between conditions more difficult. We verified this explanation by running the same analyses for the overall lower GMM scores, using the same blocks, speech material and semantic structure. When comparing the flat and compositional frame structures, we found a considerable average improvement of 19% after one training block, $t_{(74)} = 9.8, p < 0.001$, and 7% for the maximal number of training blocks, $t_{(74)} = 3.6, p < 0.001$.

We probably obtain a good performance using a flat semantic structure, because the NMF-based acoustic representation is sufficiently distinctive to set each command apart. As a consequence, the more elaborated semantic frame structure becomes redundant. However, when the GMM-based processing flow provides less distinctive representations, information contained in the semantic frame structure becomes vital to the decision process. Nevertheless, overall results are in favour of the hierarchical approach confirming our hypothesis that using additional knowledge in the form of a hierarchical semantic frame structure is an effective method to boost performance.

5.9 Conclusion and future work

This work presents results on the recently recorded dysarthric-speech database DOMOTICA-3, with speech intelligibility ranging from normal to sever dysarthric

levels. Our NMF-based framework yields 90% to 100% F-score for all speakers, with typically 70% F-score after a single example. These scores validate the use of NMF-based learning as the basis for a self-taught vocal interface for normal and dysarthric speech.

The results on PATCOR and DOMOTICA-3 demonstrate higher asymptotic F-scores by using a more advanced semantic frame structure. The lower scores for the patience card game players, using words like 'red' and 'black' instead of anticipated semantic suit concepts, further confirm the importance of using a semantic structure with more levels similar to categories used in humans. However, the mismatch in users concepts and the concepts that designers had in mind in their applications is considered a weak aspect in our framework in spite of its overall strength. Therefore, we will focus on generic procedures in future work to induce a proper semantic structure. Moreover, further improvements are expected from embedding an algorithm to detect synonyms as alternative referents to the device slot values.

The hierarchical semantic frame structure was superimposed by a decision process dominating decoded NMF activations. This decision stage can be integrated into the NMF procedure by using group sparsity [32] which obviates the need for a back-end decision stage in future work. All these moderations will boost performance, bringing us one step further in the design process towards a self-taught non-standard speech interface.

5.10 References

[1] G. Zon and M. Roerdink, "Using voice to control the civil flightdeck," Tech. Rep. NLR-TP-2006-720, National Aerospace Laboratory, 2007. pages 119

[2] R. Kuhn, J.-C. Junqua, P. Nguyen, and N. Niedzielski, "Rapid speaker adaptation in eigenvoice space," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 6, pp. 695–707, 2000. pages 119

[3] G. Potamianos and C. Neti, "Automatic speechreading of impaired speech," in *AVSP 2001-International Conference on Auditory-Visual Speech Processing*, (Volterra, Italy), pp. pp.177 –182, 2001. pages

[4] F. Rudzicz, "Acoustic transformations to improve the intelligibility of dysarthric speech," in *Proc SLPAT*, (Edinburgh, Scotland), pp. 11–21, Association for Computational Linguistics, 2011. pages 119

[5] P. Green, J. Carmichael, A. Hatzis, P. Enderby, M. S. Hawley, and M. Parker, "Automatic speech recognition with sparse training data for

dysarthric speakers.," in *Proc Interspeech*, (Geneva, Switzerland), pp. 1189–1192, 2003. pages 119

[6] M. Parker, S. Cunningham, P. Enderby, M. Hawley, and P. Green, "Automatic speech recognition and training for severely dysarthric users of assistive technology: the stardust project," *Clinical linguistics & phonetics*, vol. 20, no. 2-3, pp. 149–156, 2006. pages

[7] A. M. Acrey, *Speech recognition in individuals with dysarthria*. PhD thesis, Texas Tech University, 2012. pages 119

[8] S.-O. Caballero-Morales, "Estimation of phoneme-specific hmm topologies for the automatic recognition of dysarthric speech," *Computational and Mathematical Methods in Medicine*, vol. 2013, 2013. pages 119

[9] S.-O. Caballero-Morales and F. Trujillo-Romero, "Evolutionary approach for integration of multiple pronunciation patterns for enhancement of dysarthric speech recognition," *Expert Systems with Applications*, vol. 41, no. 3, pp. 841–852, 2014. pages 119

[10] Y. Hwang, D. Shin, C.-Y. Yang, S.-Y. Lee, J. Kim, B. Kong, J. Chung, S. Kim, and M. Chung, "Developing a voice user interface with improved usability for people with dysarthria," in *Computers Helping People with Special Needs* (K. Miesenberger, J. Klaus, W. Zagler, and A. Karshmer, eds.), vol. 7383 of *Lecture Notes in Computer Science*, pp. 117–124, Berlin Heidelberg: Springer, 2012. pages 119

[11] P. Raghavendra, E. Rosengren, and S. Hunnicutt, "An investigation of different degrees of dysarthric speech as input to speaker-adaptive and speaker-dependent recognition systems," *Augmentative and Alternative Communication*, vol. 17, no. 4, pp. 265–275, 2001. pages 119

[12] F. Rudzicz, "Comparing speaker-dependent and speaker-adaptive acoustic models for recognizing dysarthric speech," in *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility*, Assets '07, (New York, NY, USA), pp. 255–256, ACM, 2007. pages 119

[13] M. Matassoni, R. Astudillo, A. Natsamanis, and M. Ravanelli, "The dirha-grid corpus: baseline and tools for multi-room distant speech recognition using distributed microphones," in *Proc. Interspeech*, (Singapore), pp. 1613–1317, 2014. pages 120

[14] B. Lecouteux, M. Vacher, and F. Portet, "Distant speech recognition in a smart home: Comparison of several multisource asrs in realistic conditions," *Proc Interspeech*, pp. 2273–2276, 2011. pages 120

[15] H. Christensen, I. Casanuevo, S. Cunningham, P. Green, and T. Hain, "homeservice: Voice-enabled assistive technology in the home using cloud-based automatic speech recognition," in *Proc SLPAT*, (Grenoble, France), pp. 29–34, 2013. pages 120

[16] J. Gemmeke, B. Ons, M. Tessema, J. van de Loo, G. De Pauw, W. Daelemans, J. Huyghe, J. Derboven, L. Vuegen, B. Van Den Broeck, and H. Van hamme, "Self-taught assistive vocal interfaces: An overview of the aladin project," in *Proc Interspeech*, (Lyon, France), pp. 2038–2043, 2013. pages 120

[17] J. Driesen, *Discovering words in speech using matrix factorization.* PhD thesis, K.U.Leuven, ESAT, July 2012. pages 120, 124

[18] B. Ons, N. Tessema, J. van de Loo, F. Gemmeke, Jort, G. De Pauw, W. Daelemans, and H. Van hamme, "A self learning vocal interface for speech-impaired users," in *Proc SLPAT*, (Grenoble, France), pp. 1–9, 2013. pages 120, 131, 132, 133, 135, 136, 137

[19] Y. Wang and A. Acero, "Rapid development of spoken language understanding grammars," *Speech Communication*, vol. 48, no. 3-4, pp. 390–416, 2006. pages 123

[20] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *In IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980. pages 124

[21] B. Ons, J. F. Gemmeke, and H. Van hamme, "Fast vocabulary acquisition in an NMF-based self-learning vocal user interface," *Computer Speech & Language*, vol. 28, no. 4, pp. 997–1017, 2014. pages 124

[22] H. Van hamme, "Hac-models: a novel approach to continuous speech recognition," in *Proc. Interspeech*, (Brisbane, Australia), pp. 255–258, 2008. pages 125, 126

[23] M. Van Segbroeck and H. Van hamme, "Unsupervised learning of time-frequency patches as a noise-robust representation of speech," *Speech Communication*, vol. 51, pp. 1124–1138, 2009. pages 125

[24] A. Cichocki, R. Zdunek, A.-H. Phan, and S. Amari, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation.* Ltd: John Wiley & Sons, 2009. pages 125

[25] Z. Akata, C. Thurau, and C. Bauckhage, "Non-negative matrix factorization in multimodality data for segmentation and label prediction," in *16th Computer Vision Winter Workshop*, 2011. pages 126

[26] H. Lee, J. Yoo, and S. Choi, "Semi-supervised nonnegative matrix factorization," *Signal Processing Letters, IEEE*, vol. 17, no. 1, pp. 4–7, 2010. pages 126

[27] D. Lee and H. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999. pages 126

[28] B. Lize, D. Katrien, G. Jort F, and H. Van hamme, "Comparing and combining classifiers for self-taught vocal interfaces," in *Proc SLPAT*, (Grenoble, France), pp. 21–28, 2013. pages 131

[29] C. Middag, *Automatic Analysis of Pathological Speech*. PhD thesis, Ghent University, Belgium, 2012. pages 133

[30] W. S. Cleveland, "Robust locally weighted regression and smoothing scatterplots," *Journal of the American statistical association*, vol. 74, no. 368, pp. 829–836, 1979. pages 138

[31] W. S. Cleveland and S. J. Devlin, "Locally weighted regression: an approach to regression analysis by local fitting," *Journal of the American Statistical Association*, vol. 83, no. 403, pp. 596–610, 1988. pages 138

[32] R. Jaiswal, D. Fitzgerald, E. Coyle, and S. Rickard, "Shifted NMF with group sparsity for clustering NMF basis functions," in *Proc at 15th International Conference on Digital Audio Effects DAFx-12*, (York, UK), pp. 17–21, 2012. pages 144

Chapter 6

Incremental adaptive learning in the self-taught vocal interface

This chapter is based on the following article:
B. Ons, J.F., Gemmeke, and H. Van hamme (2015), "Adaptive Incremental Learning in a Self-taught Vocal User Interface," IEEE/ACM Transactions on Audio, Speech and Language Processing (submitted).

6.1 Abstract

Automatic Speech Recognition (ASR) systems are typically set up with Hidden Markov Models (HMM). The State of the art ASR models are trained on large amounts of recorded speech data and benefit from the availability of annotated speech material. Occasionally, adaptation procedures are integrated to provide speaker-adaptive ASR. However, this approach falls short when used for non-standard speech such as dysarthric speech, or when used for applications for which the interaction protocols are difficult to define beforehand. Speech technology would benefit from training during usage; adapting to the specific vocalizations and emerging expressions of the end user. We propose a vocal user interface (VUI) model that is able to learn speech recognition and understanding from demonstrations during usage. The VUI learns the acoustic representation of semantic concepts incrementally and adapts online to changes in pronunciations or word usage. The representations are learned by using non-negative matrix factorization (NMF) and the acoustic features are based on a Gaussian mixture model (GMM) that unfolds during usage. These online learning procedures are all based on Maximum A Posteriori (MAP) estimation. In a series of experiments, we compare them with their batch learning variants and demonstrate competitive learning rates and a superior adaptive capacity by incorporating a forgetting factor.

6.2 Context and contributions of the chapter

The main contribution is a new VUI model that learns online from scratch. For this accomplishment, we took several steps. First we adopted an incremental MAP algorithm for estimating GMMs. Second, we introduced a forgetting factor in the MAP-based GMM model. Third, we adopted the MAP-based incremental NMF approach from [1]. Fourth, we adapted the normalization towards a stream-based normalization, i.e. separate normalization for each parallel acoustic stream and the semantic stream. Finally, we introduced a transformation that adapts the GMM-based NMF representations to the online developing GMM. We demonstrate that this last measure induces an improvement of approximately 9% absolute after 100 training examples. Another contribution is that we demonstrate adaptation. One more upside of the new VUI model is its limited use of memory for storing learning examples. All these measures result in an incremental, adaptive, memoryless and fast learning VUI model.

6.3 Introduction

Automatic Speech Recognition (ASR) systems are typically set up with Hidden Markov Models (HMM), developed with continuous Gaussian mixture (GMM) emission densities and context-dependent phones. Currently, Deep Neural Networks (DNN) that have many hidden layers outperform GMM's on a variety of speech recognition benchmarks [2]. These state of the art ASR systems are trained on large amounts of recorded speech data and benefit from the availability of annotated speech material. The amounts that are required to build a competitive ASR system are usually available for widely spoken languages and for large-scale applications with great economical potential such as speech-to-speech and speech-to-text translation. However, the majority of languages are low-resource languages with a lot of peculiarities in phonotactics, word segmentation or morphology, or dialects lacking strict language convention. Moreover, a considerable share of currently developed ASR applications are tailored solutions potential developed for one customer only or for a small user group.

It is in this broader scope that we are developing a vocal user interface (VUI) for non-standard speech in low resource settings, that is, with a few utterances of training data per command (for an overview of the ALADIN VUI, see [3]). The system does not require word segmentation and benefits from rather abstract supervision such as utterance-based semantic content. This kind of supervision unfolds naturally by mining the VUI usage and by automating VUI interactions in which the user is asked to give demonstrations of his spoken commands, choosing his own words. We aim to build a system that learns speech recognition from scratch at deployment, thus offering viable speech recognition solutions for small-vocabulary, small-user group applications such as voice-enabled home automation and voice-driven assistive aids targeting users with non-standard speech. Typical user groups are elderly or people with dysarthria.

The user's speech and the action states in the command-and-control application are two sources of information that we combine by using Non-negative Matrix Factorisation (NMF, see [4]). This machine learning procedure allows the VUI to learn recurrent co-occurring patterns in the semantic and acoustic input. These patterns pertain to the user-specific vocabulary. In [5], it was demonstrated that this procedure learns from a few demonstrations if model-based statistical features are used such as co-occurrence statistics of GMM posteriors or HMM phone posteriors. Moreover, in a comparative study [6] with conventional ASR methods adapted to dysarthric speech (see the STARDUST [7] and VIVOCA [8] projects), it was shown that the NMF-based system provides competitive results in word and sentence-based recognition accuracy, but offers a substantial reduction in the training material needed to approach asymptotic accuracy.

Another fast learning algorithm operating on limited storage space and small vocabulary is Dynamic Time Warping (DTW) [9, 10]. DTW is a template-based technology using a dynamic programming alignment process to find the similarity between two speech signals. In this study, we aim to unify model-based advances such as model adaptation with template-based advantages such as fast speaker-dependent learning and the use of limited storage resources. Whereas the NMF-based approach has been compared with conventional HMM and GMM methods [6, 11, 12], we incorporate a DTW baseline in this study. Although DTW is an early developed ASR technique, DTW has been popular in lots of applications despite its limitations with respect to adaptation and robustness. For example, in [9] a HMM-like DTW procedure was proposed in which HMM-like acoustic models were trained for each of DTW referenced templates. Their procedure enables model adaptation and merges different word examples in one template. Inspired by [13], we introduce an adaptive DTW procedure by updating the DTW referenced templates by the last online presented examples.

Voice-enabled assistive technology for the physically incapacitated is investigated in projects such as DIRHA [14], SWEET-HOME [15] and HomeService [16]. Speaker-independent ASR systems are used together with speaker adaptation procedures. Contrary to the adaptation approach, the targeted VUI training procedure is aimed at building semantic-acoustic representations from online learning using speech and demonstrations of the user. A typical aspect of the training material consisting of interactive experiences is the incremental data exposure of user commands and demonstrations. One of the main contributions in this study is the fitting of Maximum A Posteriori (MAP) algorithms into incremental learning procedures operating on weak supervision and incrementally exposed speech data. To this end, we adapted probabilistic incremental models as the alternative version of the batch learning procedures pursued in the preceding studies [5, 6] and pursued adaptivity by incorporating a forgetting factor in the incremental models. Similar to the DTW approach that does not require model training, our VUI model is, to the best of our knowledge, the first model-based approach that builds its ASR models from scratch, that is from preprocessed features such as MFCC features and utterance-based semantic content. In earlier work, the VUI model used batch learning procedures that required data storage and computational resources that correlated with the amounts of stored training data. Oposed to batch learning, the introduced VUI model in this chapter does not store data and uses limited computational resources, as processing only involves the commands of the current actions. Another contribution of the chapter is the empirical comparison between incremental and batch learning procedures considering real learning environments targeting Command and Control (C&C) home automation for dysarthric speech. These experiments focus on fast learning and

life-span adaptation to user's vocal characteristics.

The remainder of the chapter is organised as follows. In Section 6.4, we describe NMF batch learning as it was implemented in preceding studies. In Section 6.5, we adapt existing MAP algorithms to the demonstration-driven incremental learning context. Based on these algorithms, we compose several realistic procedures in Section 6.6 that we validate in a series of experiments reported in Section 6.7. These experiments aim at fast learning and adaptation. Then, we discuss the feasibility of our approach, the effectiveness of MAP incremental procedures and our thoughts on future work in Section 6.8. Our conclusion is presented in Section 6.9.

6.4 The vocal user interface: preliminaries

In the following, let $\mathbf{U}_n$ denote the n^{th} utterance. Each spoken utterance is composed of a sequence of frame vectors: $\mathbf{U}_n = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(t)}, \ldots, \mathbf{x}^{(T_n)}]$, where $\mathbf{x}^{(t)}$ consists of a column-wise feature vector such as Mel-frequency cepstral coefficients (MFCC's), commonly used as features in speech recognition systems. The incremental index t follows the sequential order of the frames.

The acoustic feature vectors in the VUI, proposed in [5] and [12], are built in two layers: a clustering and a factorization layer. In the first layer, a GMM with K components is used to transform the feature vectors in $\mathbf{U}_n$ into a posteriorgram. A posteriorgram is a matrix expressing the posterior probability that a frame at time t is generated by the k^{th} Gaussian, denoted by f_k. If $k = 1, \ldots, K$ and $t = 1, \ldots, T_n$, then the utterance-based posteriorgram is of size $k \times T_n$.

In the second layer, the data is factorised and for this, fixed-length vectors are required. Therefore, posterior likelihoods are converted into Histogram of Acoustic Co-occurrence (HAC) features (see [17]) by accumulating the probability of observing a frame at time t and another frame at time $t + \tau$ generated by the Gaussian components f_k and f_l, respectively, with $1 \leq k, l \leq K$ and t proceeding from 1 to $T_n - \tau$. The accumulated scores for all $K \times K$ co-occurring Gaussian pairs in utterance n are stacked in a column vector denoted by $\mathbf{v}_n$. If the number of Gaussian mixture components is held constant, then all utterance-based feature vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{n-1}$ have the same length. The matrix composed of all utterance-based HAC features including utterance n and its preceding utterances is denoted by $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n]$.

The utterance-based column vector $\mathbf{v}_n$ is augmented with a binary column vector $\mathbf{a}_n$, representing the relevant semantics that users refer to when they control a device by voice. For this, all concepts that describe C&C actions in the VUI-user context are predefined and a fixed-length vector is composed in

referent <Kitchen door>	0	1	0	0
referent <open>	1	1	0	1
referent <living room>	0	0	0	1
referent <blinds>	1	0	0	1
Gaussians: $1 \prec 1$	0.1	0	0.1	0
Gaussians: $1 \prec 2$	2.5	0	0.8	0.1
Gaussians: $1 \prec 3$	0.5	0	0	4
Gaussians: $2 \prec 1$	0	0	0	0.5
Gaussians: $2 \prec 2$	0	0	2	0
Gaussians: $2 \prec 3$	0	0	0	0
Gaussians: $3 \prec 1$	0	0.9	0	0
Gaussians: $3 \prec 2$	0	1	0	0
Gaussians: $3 \prec 3$	0.2	0	0	0

Table 6.1: *Example of a data matrix with four semantic entries and HAC features for three Gaussians*

which entries represent the presence or absence of a vocal expression referring to one of these predefined semantics. The presence or absence of a spoken referent for these predefined semantics is brought in by mining the demonstrated action on the targeted devices. The collection of this information is application- and device-dependent. For the purpose of giving a general VUI description, we assume here that this information is given and presented in a binary vector for each spoken utterance. The collection of all semantics including those guiding utterance n, is denoted by $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$. In Table 6.1, a data matrix with four columns is depicted and each column represents one utterance. The first utterance is an expression in which the user demonstrated the opening of the blinds. This action is guided with nine acoustic co-occurrence scores in this toy example. While the upper part in Table 6.1 exemplifies the $\mathbf{A}$ matrix, the lower part exemplifies the $\mathbf{V}$ matrix.

Users can choose their own words. Therefore, a machine learning procedure is required that is able to learn from semantic supervision without annotations in terms of specific word usage. Moreover, as can be seen in Table 6.1, supervision does not include word order, segmentation markers or phonetic descriptions. It was shown in [5] and [12] that NMF is able to fulfill these requirements. Semantic and acoustic input data were jointly factorized in order to find the HAC patterns that co-occur with the semantic entities:

$$\begin{bmatrix} \mathbf{A} \\ \mathbf{V} \end{bmatrix} \approx \begin{bmatrix} \mathbf{W}_0 \\ \mathbf{W}_1 \end{bmatrix} \mathbf{H}. \tag{6.1}$$

The co-occurrence of semantic and acoustic features are found in the columns of $\mathbf{W}_0$ and $\mathbf{W}_1$, respectively, whereas the columns in $\mathbf{H}$ indicate which co-occurring patterns are active in the respective utterance-based columns in $\mathbf{A}$ and $\mathbf{V}$.

The non-negative matrix factorization is obtained by minimizing the Kullback-Leibler divergence between both sides, so that

$$(\mathbf{H}, \mathbf{W}_1, \mathbf{W}_0) = \underset{(\mathbf{H}^*, \mathbf{W}_1^*, \mathbf{W}_0^*)}{\arg\min} \left[D_{KL}\left(\mathbf{V} || \mathbf{W}_1^* \mathbf{H}^*\right) + \beta D_{KL}\left(\mathbf{A} || \mathbf{W}_0^* \mathbf{H}^*\right) \right] \quad (6.2)$$

with β a weight balancing the relative importance of co-occurring semantic-acoustic patterns against the recurrence of acoustic data patterns. A common practice is to match the L1-norm of $\mathbf{A}$ and $\mathbf{V}$, and set β equal to one.

6.5 Incremental learning

We propose a VUI model that adopts the global structure in [5] and [12] — which consists of a clustering layer and a factorization layer — but performs incremental learning. In the clustering layer, a GMM is trained incrementally. The GMM transforms feature vectors $\mathbf{x}^t$ into a posteriorgram. In the factorization layer, incremental NMF learning [18] associates the HAC features in $\mathbf{v}_n$ to the semantics in $\mathbf{a}_n$. Incremental NMF is closely related to Probabilistic Latent Semantic Analysis (PLSA), which can be thought of to consist of the probabilistic version of NMF with the Kullback-Leibler divergence as cost function (see [1, 19, 20]).

For incremental learning, the method of maximum a posteriori (MAP) estimation is adopted. In the following, MAP estimation is explained for GMM's [21] and PLSA [22], then, we transpose the PLSA method to incremental NMF and include a forgetting factor in both layers. Since both layers learn from scratch, we explain how changes in the clustering layer are treated in the factorization layer.

6.5.1 MAP estimation

Suppose that input data is available in chunks presented in separate and sequential epochs. The sequential order is denoted by the index i. Each epoch contains a number of utterances denoted by the constant O_i. Presume that utterance n is the last utterance in epoch i and that all utterances

in i are contained in a matrix denoted by $\mathbf{U}^{(i)}$, then $n = \sum_{j=1}^{i} O_j$ and $\mathbf{U}^{(i)} = [\mathbf{U}_{n-O_i+1}, \ldots, \mathbf{U}_{n-1}, \mathbf{U}_n]$. Lets denote all input data from all preceding epochs by $\mathcal{U}^{(i-1)} = \{\mathbf{U}^1, \ldots, \mathbf{U}^{(i-2)}, \mathbf{U}^{(i-1)}\}$. Similarly, the utterance-based feature vectors are presented epoch-wise as follows: $\mathbf{V}^{(i)} = [\mathbf{v}_{n-O_i+1}, \ldots, \mathbf{v}_{n-1}, \mathbf{v}_n]$ and $\mathbf{A}^{(i)} = [\mathbf{a}_{n-O_i+1}, \ldots, \mathbf{a}_{n-1}, \mathbf{a}_n]$. The data set in all preceding epochs is represented by $\mathcal{V}^{(i-1)} = \{\mathbf{V}^{(1)}, \ldots, \mathbf{V}^{(i-2)}, \mathbf{V}^{(i-1)}\}$ and $\mathcal{A}^{(i-1)} = \{\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(i-2)}, \mathbf{A}^{(i-1)}\}$.

The following equation expresses the relation between the prior and the posterior distribution of the parameter set for the incremental GMM.

$$g(\theta|\mathcal{U}^{(i)}) \propto f(\mathbf{U}^{(i)}|\theta) g(\theta|\mathcal{U}^{(i-1)}) \tag{6.3}$$

with θ denoting the GMM parameter set, g denoting the joint probability of the parameter set given the prior exposed data and f denoting the likelihood of the data in epoch i, given the parameter set. The mode of the posterior distribution is defined as follows

$$\theta_{MAP} = \arg\max_{\theta} f(\mathbf{U}^{(i)}|\theta) g(\theta|\mathcal{U}^{(i-1)}) \tag{6.4}$$

If we consider HAC features and semantics separately, then MAP estimates in both streams are defined as (see [22])

$$\varphi_{MAP} = \arg\max_{\varphi} f(\mathbf{V}^{(i)}|\varphi) g(\varphi|\mathcal{V}^{(i-1)}) \tag{6.5}$$

$$\vartheta_{MAP} = \arg\max_{\vartheta} f(\mathbf{A}^{(i)}|\vartheta) g(\vartheta|\mathcal{A}^{(n-1)}) \tag{6.6}$$

with φ and ϑ the parameter set of the PLSA model for the HAC features and the semantics, respectively.

MAP estimation is less complicated if f is chosen from the exponential family and g from the respective conjugate family. This combination possesses a sufficient statistic of fixed dimension, meaning that the parameters only depend on the data through the sufficient statistics. Consequently, all relevant information for parameter estimation is passed on to the following epoch by keeping track of a few data-dependent statistics, thus obviating the need for storing data.

6.5.2 MAP updates in the GMM

If the total number of frames in epoch i is $T = \sum_{j=n-O_i}^{n} T_j$, then the likelihood function of the GMM with K p-dimensional multivariate normal densities is

expressed as follows

$$f(\mathbf{U}^{(i)}|\theta) = \prod_{t=1}^{T}\sum_{k=1}^{K}\omega_k f_k(\mathbf{x}_t|\mu_k, \Sigma_k). \tag{6.7}$$

where ω_k denotes the mixture proportion for the k^{th} mixture component subject to $\sum_{k=1}^{K}\omega_k = 1$ and $f_k \sim \mathcal{N}(\mu_k, \Sigma_k)$.

Unfortunately, the probability density function (p.d.f.) of a GMM is not a member of the exponential family. Moreover, the mixture component generating the observation is unknown. The expectation-maximization (EM) [23] algorithm is often used in this case when models involve incomplete data. The EM algorithm exploits the fact that the complete-data likelihood is easier to maximize than the likelihood of the incomplete data. The complete data likelihood is the joint likelihood of the observed data and the missing data pertaining to the occupation of observations in the mixture components. The idea in [21] was to consider the generative process of the complete data as being modelled by the joint p.d.f. of two distributions from the exponential family. This implementation still allows for an easy updating scheme. Gauvain and Lee [21] proposed a multinomial distribution for the sample sizes of the component distributions and a multivariate Gaussian density for each component population. They assumed a Dirichlet distribution for the prior density of the multinomial parameters. These parameters correspond with the mixture proportions ω_k of the GMM,

$$g(\omega_1, \omega_2, \ldots, \omega_K|\alpha_1, \alpha_2, \ldots, \alpha_K) \propto \prod_{k=1}^{K}\omega_k^{\alpha_k - 1}, \tag{6.8}$$

where $\alpha_k > 0$ are parameters of the Dirichlet distribution. Gauvain and Lee used a normal Wishart density as the conjugate prior for the precision matrix. Equivalently, we use the normal-inverse Wishart as the conjugate prior for the variance-covariance matrix Σ_k. The normal-inverse Wishart takes the form

$$\begin{aligned} g(\mu_k, \mathbf{\Sigma}_k|\mu_{0k}, \lambda_k, \mathbf{\Psi}_k, \nu_k) &\propto \frac{1}{|\mathbf{\Sigma}_k|^{\frac{\nu_k + p + 1}{2}}} \\ exp[(-\frac{\lambda_k}{2}(\mu_k - \mu_{0k})^T\mathbf{\Sigma}_k^{-1}(\mu_k - \mu_{0k})]exp(-\frac{1}{2}tr(\mathbf{\Psi}_k\mathbf{\Sigma}_k^{-1})), \end{aligned} \tag{6.9}$$

where $(\mu_{0k}, \lambda_k, \mathbf{\Psi}_k, \nu_k)$ are hyper parameters such that $\lambda_k > 0$ and $\nu_k > p - 1$. The total prior density is the product of the prior in Eq. 6.8 and 6.9:

$$g\big(\theta|\mathcal{U}^{(i-1)}\big) = g(\omega_1, \omega_2, \ldots, \omega_K)\prod_{k=1}^{K} g(\mu_k, \mathbf{\Sigma}_k). \tag{6.10}$$

MAP estimates in [21] are obtained by using the EM algorithm [23]. The algorithm consists of iteratively maximizing the auxiliary function $\mathcal{R}(\hat{\theta}, \theta)$ which is composed of two terms:

$$\mathcal{R}(\hat{\theta}, \theta) = Q(\hat{\theta}, \theta) + \log\big(g\big(\theta|\mathcal{U}^{(i-1)}\big)\big). \tag{6.11}$$

$Q(\hat{\theta}, \theta)$ is the auxiliary function used to obtain ML estimates and $\hat{\theta}$ denotes the MAP and the ML estimate of θ using R and Q, respectively. Organising the exponential of $e^{\mathcal{R}(\hat{\theta},\theta)}$ in the same form as its prior in Eq. 6.10 yields the following equations [21]:

$$c_{kt}^{(i)} = \frac{\hat{\omega_k}^{(i)} f_k(\mathbf{x}_t|\hat{\mu_k}^{(i)}, \hat{\mathbf{\Sigma}}_{\mathbf{k}}^{(i)})}{\sum_{k=1}^{K} \hat{\omega_k}^{(i)} f_k(\mathbf{x}_t|\hat{\mu_k}^{(i)}, \hat{\mathbf{\Sigma}}_{\mathbf{k}}^{(i)})}. \tag{6.12}$$

with $c_{kt}^{(i)}$ the posterior likelihood that sample $\mathbf{x}_t$ is generated by Gaussian k. The occupation number for component k, denoted by $c_k^{(i)}$, is given by

$$c_k^{(i)} = \sum_{t=1}^{T_i} c_{kt}^{(i)}. \tag{6.13}$$

The following statistics are adjusted in each EM step, and updated after convergence for each new epoch i:

$$\alpha_k^{(i)} = \alpha_k^{(i-1)} + c_k^{(i)}, \tag{6.14}$$

$$\nu_k^{(i)} = \nu_k^{(i-1)} + c_k^{(i)}, \tag{6.15}$$

$$\lambda_k^{(i)} = \lambda_k^{(i-1)} + c_k^{(i)}, \tag{6.16}$$

$$\mathbf{X}_k^{(i)} = \mathbf{X}_k^{(i-1)} + \sum_{t=1}^{T_i} c_{kt}^{(i)} \mathbf{x}_t, \tag{6.17}$$

$$\mathbf{S}_k^{(i)} = \mathbf{S}_k^{(i-1)} + \sum_{t=1}^{T_i} c_{kt}^{(i)} \mathbf{x}_t \mathbf{x}_t'. \tag{6.18}$$

These statistics are used to obtain the MAP parameters in each maximization step as follows,

$$\hat{\omega}_k^{(i)} = \frac{\alpha_k^{(i)} - 1}{\sum_{j=1}^{K} \alpha_j^{(i)} - K}, \quad \alpha_k > 1, \tag{6.19}$$

$$\hat{\mu}_k^{(i)} = \frac{\mathbf{X}_k^{(i)}}{\lambda_k^{(i)}}, \tag{6.20}$$

$$\hat{\mathbf{\Sigma}}_k^{(i)} = \frac{\mathbf{S}_k^{(i)} - \frac{\mathbf{X}_k^{(i)} \mathbf{X}_k^{(i)'}}{\lambda_k^{(i)}}}{\nu_k^{(i)} + p + 1}. \tag{6.21}$$

Note that the notation and equations differs from those in [21] where MAP updates, but no incremental learning was introduced.

6.5.3 MAP updates in PLSA

PLSA [24] is used in search engines where the co-occurrence of words and documents is explained by a latent topic variable. PLSA is a model of the observed joint probability of two discrete variables. The joint probability is modelled as a mixture of conditionally independent multinomial distributions, given a latent variable. We denote the co-occurring variables by $m_f \in \mathbf{M} = \{m_1, m_2, \ldots, m_F\}$ representing the occurrence of an acoustic event that increments the f_{th} entry in $\mathbf{v}_n$ with one and $d_n \in \mathbf{D} = \{d_1, d_2, \ldots, d_N\}$ representing the occurrence of utterance n. We denote the latent variable by $z_j \in \mathbf{Z} = \{z_1, z_2, \ldots, z_J\}$ representing the occurrence of a latent entity underlying the occurrence of v_{fn} in utterance n. The joint probability of the observed pair (m_f, d_n) depends on $\mathbf{Z}$ as follows [24]:

$$P(m_f, d_n) = P(d_n) \sum_{j=1}^{J} P(m_f|z_j) P(z_j|d_n) \tag{6.22}$$

If HAC feature v_{fu} represents the number of events for the co-occurrence of m_f in utterance d_u with u an utterance indicator for the current epoch i, then the likelihood of the data in epoch i is proportional to,

$$f(\mathbf{V}^{(i)}|\varphi) \propto \prod_{f=1}^{F} \prod_{u=n-O_i}^{n} P(m_f, d_u)^{v_{fu}} \tag{6.23}$$

with φ denoting the parameter vector containing $P(m_f|z_j)$ and $P(z_j|d_u)$. The parameter vector containing $P(d_u)$ is trivially found by marginalizing $P(m_f, d_u)$ over m_f.

In [1] and [21], the joint prior p.d.f. of the parameter vector was chosen to consist of Dirichlet distributions. The prior density is specified as

$$g(\varphi|\mathcal{V}^{(i-1)}) = \prod_{j=1}^{J} \left(\prod_{f=1}^{F} P(m_f|z_j)^{\xi_{fj}-1} \right), \tag{6.24}$$

where $\xi_{fj} > 0$ are Dirichlet parameters. Note that this prior density does not include the p.d.f. on the parameter $P(z_j|d_u)$, which is a simplification justified in [1] by considering the occurrence of an utterance to carry no information. Therefore, this variable does not carry useful information to the next epoch.

The same procedure in the semantic stream yields the following proportional relation:

$$f(\mathbf{A}^{(i)}|\vartheta) \propto \prod_{r=1}^{R} \prod_{u=n-O_i}^{n} P(g_r, d_u)^{a_{ru}}, \tag{6.25}$$

with ϑ the PLSA parameter vector corresponding with the semantic stream, with R the dimension of $\mathbf{a}_u$ and with $g_r \in \mathbf{G}$ a variable representing the occurrence of a semantic event that increments the entry a_{ru} in $\mathbf{a}_u$ with one.

The prior density of the semantic variables ϑ is expressed as follows,

$$g(\vartheta|\mathcal{A}^{(i-1)}) = \prod_{j=1}^{J} \left(\prod_{r=1}^{R} P(g_r|z_j)^{\iota_{rj}-1} \right) \tag{6.26}$$

with $\iota_{rj} > 0$ composing parameters of the Dirichlet density in the semantic stream.

In [1], the auxiliary function $\mathcal{R}(\hat{\varphi}, \varphi)$ was extended with a forgetting factor γ in order to weigh recently collected data statistics heavier than previously collected statistics, thereby providing adaptation to changes in the vocabulary. Here, we incorporate the same forgetting factor and extend the auxiliary function with the likelihood of the semantic stream. We assume independence between both streams given the latent variable $\mathbf{Z}$ and optimize the following loss function,

$$\begin{aligned} \mathcal{R}(\hat{\varphi}, \varphi) + \beta\mathcal{R}(\hat{\vartheta}, \vartheta) &= Q(\hat{\varphi}, \varphi) + \beta Q(\hat{\vartheta}, \vartheta) \\ &\quad + \gamma \left(\log\left(g\big(\varphi|\mathcal{V}^{(i-1)}\big)\right) + \beta \log\left(g\big(\vartheta|\mathcal{A}^{(i-1)}\big)\right) \right). \end{aligned} \tag{6.27}$$

Considering that both streams share the same latent variable $P(z_j|d_u)$, the expectation step leads to the following equations:

$$P(z_j|m_f, d_u) = \frac{P(m_f|z_j)P(z_j|d_u)P(d_u)}{\sum_{p=1}^{J} P(m_f|z_p)P(z_p|d_u)P(d_u)}, \tag{6.28}$$

$$P(z_j|g_r, d_u) = \frac{P(g_r|z_j)P(z_j|d_u)P(d_u)}{\sum_{p=1}^{J} P(g_r|z_p)P(z_p|d_u)P(d_u)}, \tag{6.29}$$

and the following equations compose the maximization step:

$$\xi_{fj}^{(i)} = \gamma(\xi_{fj}^{(i-1)} - 1) + 1 + \sum_{u=n-o}^{n} v_{fu} P(z_j|m_f, d_u), \tag{6.30}$$

$$\iota_{rj}^{(i)} = \gamma(\iota_{rj}^{(i-1)} - 1) + 1 + \sum_{u=n-o}^{n} a_{ru} P(z_j|g_r, d_u), \tag{6.31}$$

$$P(m_f|z_j) = \frac{\xi_{fj}^{(i)} - 1}{\sum_{f=1}^{F} \xi_{fj}^{(i)} - F}, \quad \xi_{fj} > 1, \tag{6.32}$$

$$P(g_r|z_j) = \frac{\iota_{rj}^{(i)} - 1}{\sum_{r=1}^{R} \iota_{rj}^{(i)} - R}, \quad \iota_{rj} > 1, \tag{6.33}$$

$$P(z_j|d_u) = \frac{\sum_{f=1}^{F} v_{fu} P(z_j|m_f, d_u) + \beta \sum_{r=1}^{R} a_{ru} P(z_j|g_r, d_u)}{\sum_{p=1}^{K} \left(\sum_{f=1}^{F} v_{fu} P(z_p|m_f, d_u) + \beta \sum_{r=1}^{R} a_{ru} P(z_p|g_r, d_u) \right)}. \tag{6.34}$$

with β a weighting factor identical to the one in Eq. 6.2. Note that the notation differs from [1] where the updates are expressed in function of the parameter $\kappa_{fj} = \xi_{fj} - 1$ and where the semantics and acoustics are treated as one.

The above equations can be interpreted as a probabilistic version of an extension of the NMF described in Section 6.4. via the relations:

$$a_{ru} = c_u P(g_r, d_u), \qquad v_{fu} = c_u P(m_f, d_u) \tag{6.35}$$

$$w_{0,rj} = P(g_r|z_j), \qquad w_{1,fj} = P(m_f|z_j) \tag{6.36}$$

$$h_{ju} = c_u P(z_j|d_u) \tag{6.37}$$

with a, v, w_0, w_1 and h denoting entries of $\mathbf{A}^{(u)}, \mathbf{V}^{(u)}, \mathbf{W}_0, \mathbf{W}_1$ and $\mathbf{H}$, respectively (see Eq. 6.1) and c_u an utterance-based constant.

6.5.4 GMM with forgetting factor

Gaussian parameters are tuned incrementally to the user's speech by processing an increasing number of utterances. Using MAP updates without forgetting factor will strengthen priors more and more as the number of processed utterances increases, reducing thereby the impact of more recent utterances on parameter estimation. A forgetting factor will keep priors weaker thus accelerating adaptation on a continuous basis. Similar to the forgetting factor γ in Eq. 6.30, we introduce a forgetting factor, denoted by η, in the GMM. The auxiliary function in Eq. 6.11 gets the following form:

$$\mathcal{R}(\hat{\theta}, \theta) = Q(\hat{\theta}, \theta) + \eta \log \left(g\left(\theta | \mathcal{U}^{(i-1)}\right)\right), \tag{6.38}$$

and leads to the following modifications in the equations 6.14 to 6.18,

$$\alpha_k^{(i)} = \eta(\alpha_k^{(i-1)} - 1) + 1 + c_k^{(i)}, \tag{6.39}$$

$$\nu_k^{(i)} = \eta\nu_k^{(i-1)} + (\eta - 1)(p + 1) + c_k^{(i)}, \tag{6.40}$$

$$\lambda_k^{(i)} = \eta\lambda_k^{(i-1)} + c_k^{(i)}, \tag{6.41}$$

$$\mathbf{X}_k^{(i)} = \eta\mathbf{X}_k^{(i-1)} + \sum_{t=1}^{T_i} c_{kt}^{(i)} \mathbf{x}_t \quad , \tag{6.42}$$

$$\mathbf{S}_k^{(i)} = \eta\mathbf{S}_k^{(i-1)} + \sum_{t=1}^{T_i} c_{kt}^{(i)} \mathbf{x}_t \mathbf{x}_t' \quad , \tag{6.43}$$

keeping all other formalism the same. The initialization of α_k, ν_k and λ_k is explained in section 6.7.1.

The influence of γ, η on the data statistics is depicted in Figure 6.1. Here, the utterance $n = 100$ is considered the most recent utterance receiving a reference weight of 1. The curves display the relative weights of the incremental statistics that are accumulated in preceding utterances $n < 100$ using Eq. 6.39 to Eq. 6.43. It can be seen in Figure 6.1 that the relative weighting is heavily altered by forgetting factors slightly deviating from one.

6.5.5 GMM modifications

On the one hand, incremental learning of GMM parameters improves the GMM gradually by the increasing availability of the data. This is especially useful for non-standard speech for which representative data is hard to find beforehand.

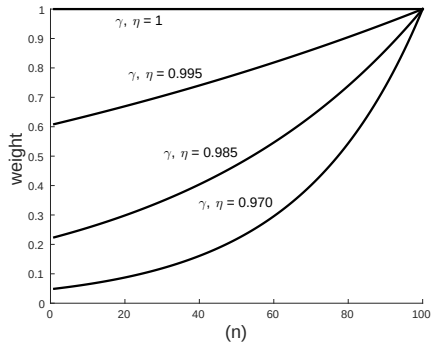


Figure 6.1: *The influence of γ, η on the relative weight of statistics collected in preceding epochs*

On the other hand, incremental learning alters Gaussian mixture components continuously, which is inopportune since these Gaussians are used as a codebook for composing HAC features. These alterations necessitates forgetting of NMF representations that are built with respect to less recent Gaussians. In addition to forgetting, i.e. weakening priors, we propose a transformation that adjusts NMF parameters directly in accordance with GMM modifications. A Gaussian component that alters its mean would induce different posteriors than the ones induced on older data. However, the NMF representations are based on past data and posteriors. One approach to adjust Gaussian alterations is to estimate how these changes would affect the posteriorgram of the data and modify the learned representations in the NMF layer accordingly.

If all data is stored, then their posteriors for the GMM estimated at epoch $i-1$ and the GMM estimated at the current epoch i are easily calculated. A $K \times K$ transformation matrix could be obtained that transforms posteriors prior to epoch i to those after epoch i. This transformation would be helpful to transform NMF-based representations to a more viable version with respect to the recent GMM. By design, data is not memorised in MAP-based incremental learning, thus impeding this approach. Therefore, we use the GMM of the data in the preceeding epoch to simulate the data. If we denote a Gaussian component estimated at epoch $i-1$ as Gaussian density function f_k and at the current epoch i as q_l, then the expected likelihood that a sample drawn from f_k is originating from a density q_l can be expressed as the exponent of the negative crossentropy. For this, we first express the log-likelihood of the simulated data

for density q_l given that the samples were drawn from density f_k,

$$\mathbb{E}_{f_k(x)}[\log l(x)] = \int_{\mathcal{R}^d} f_k(x) \log q_l(x)\, \mathrm{d}x \tag{6.44}$$

Clearly, this expression can be recognised as the negative cross entropy $-H(f_k, q_l)$ with $H(f_k, q_l)$ defined as

$$H(f_k, q_l) = \mathbb{E}_{f_k}[-\log q_l] \tag{6.45}$$

$$= H(f_k) + D_{KL}(f_k||q_l) \tag{6.46}$$

where $H(f_k)$ denote the entropy of density f_k. The negative cross entropy $-H(f_k, q_l)$ can be interpreted as the expected log-likelihood of a sample $\mathbf{x}$ considering a drawn from Gaussian q_l, but actually generated with density f_k. The closed-form for $H(f_k, q_l)$ for two Gaussian densities is

$$H(f_k, q_l) = \frac{1}{2}\big[\ln|2\pi\boldsymbol{\Sigma}_l| + tr(\boldsymbol{\Sigma}_l^{-1}\boldsymbol{\Sigma}_k) + (\mu_l - \mu_k)'\boldsymbol{\Sigma}_l^{-1}(\mu_l - \mu_k)\big] \tag{6.47}$$

Since there is no stored data, the average likelihood is used as an alternative:

$$\overline{q}_l\big(\mathbf{x}|\mu_l, \boldsymbol{\Sigma}_l, \mathbf{x} \sim \mathcal{N}(\mu_k, \boldsymbol{\Sigma}_k)\big) \sim e^{-H(f_k, q_l)} \tag{6.48}$$

The expected likelihood $\overline{q}_l$ at epoch i overlaps and the posterior likelihoods describes the expected occupation of a sample from f_k with respect to all Gaussians component densities l_j proceeding the current epoch i as follows

$$\mathbf{T}(k, l) = \frac{\overline{q}_l(\mathbf{x}|\mu_l, \boldsymbol{\Sigma}_l, \mathbf{x} \sim \mathcal{N}(\mu_k, \boldsymbol{\Sigma}_k))}{\sum_{j=1}^{K} \overline{q}_j(\mathbf{x}|\mu_{l_j}, \boldsymbol{\Sigma}_{l_j}, \mathbf{x} \sim \mathcal{N}(\mu_k, \boldsymbol{\Sigma}_k))} \tag{6.49}$$

with $\mathbf{T}$ having dimensions $K \times K$. The rows of $\mathbf{T}$ can be conceived as the repartition of the data generated by the old Gaussians into the new Gaussians. The column-wise HAC representations in $\mathbf{W}_1$ are then reshaped into square $K \times K$ matrices with accumulated co-occurring scores for all $K \times K$ Gaussian pairs, followed by left and right multiplication of $\mathbf{T}$ and its transpose, respectively. $\mathbf{T}$ could also be considered a smoother, smoothing the posteriorgram with respect to similarity between Gaussian components. It was shown in **chapter 4**, that smoothing of posteriors yields better performance of NMF-based learning from scarce data.

Nonetheless, this transformation is only useful for initial guessing of $\mathbf{W}_1$ parameters because this procedure takes only marginal changes in Gaussian-based pairwise co-occurrences into account. Therefore, new data is required to fine-tune this initial guess to real co-occurrence statistics.

6.6 Overview of the different procedures

In the preceding section, incremental VUI learning is introduced in two layers: the clustering layer and the factorization layer. The alternative to incremental learning in the clustering layer is the use of a fixed codebook. A fixed codebook has the advantage that the codebook is consistent throughout the whole experiment. Procedures based on a fixed codebook were used in [5] and [12] and briefly explained in Section 6.4. A speaker-independent codebook is acquired by applying the k-means procedure using randomly selected frames from a Dutch non-dysarhtric speech corpus. We referred to it as *CGN Fixed Codebook* ***(CGN-FC)***. After applying the k-means algorithm, full covariance Gaussians are estimated on the partition of the samples. As for the Gaussians of the GMM, these Gaussians are used to transform feature vectors into a posteriorgram.

A second alternative is to use a speaker-dependent fixed codebook by implementation of the k-means algorithm on prior recordings of the user. Although this assumes a speaker-dependent recording step, speaker-dependent training using limited amounts of available data was favoured above speaker-independent codebooks in [5]. We refer to this procedure with the term *Speaker-Dependent Fixed Codebook* ***(SD-FC)*** and use the DOMOTICA-3-precursor, namely DOMOTICA-2 (see Section 6.7.1) which contains recordings of the same speakers, for this purpose. The fixed codebooks are compared against the adaptive incremental procedure explained in Section 6.5.2. The adaptive learning procedure is referred to as *adaptive incremental GMM* ***(AI-GMM)***.

In the factorization layer, we compare *Batch NMF learning* ***(B-NMF)*** explained in Section 6.4 with the *adaptive incremental NMF* ***(AI-NMF)*** variant explained in Section 6.5.3. In batch learning, the training sets are encoded and factorized as a whole. A transformation like the one proposed in Eq. 6.49 is not required since the same codebook is used for all utterances. Nevertheless, when the number of spoken commands increases, batch learning will require more and more data memory. Contrarily to batch learning, incremental learning is memoryless in the sense that only the last data epoch is processed, and thus, memory requirements for this do not grow.

The VUI procedures are compared with *Dynamic Time Warping* ***(DTW)***, frequently used in speaker-dependent small vocabulary embedded applications. In the DTW procedure, a dynamic programming alignment process operating on local dissimilarity is used to find the global dissimilarity between two sequences of feature vectors. When comparing DTW with NMF procedures, DTW has a disadvantage with regard to the kind of supervision used in the VUI model. There are no word segmentations available and since a DTW-based template matching system does not look for recurrent data pattern, commands are learned in one piece. Contrarily, joint NMF as machine learning procedure

is capable of finding the word constituents of the utterances based on the statistical regularities; thus, it does not need word segmentations. For example, if the semantic constituents of the commands such as "Open the blinds" and "close the kitchen door" are learned, then an unseen command such as "close the blinds" is theoretically recognizable in the NMF-based decoder, but not in a DTW-based decoder. Since DTW is known as a computational expensive algorithm, only a few examples of each command are usually kept as templates. Here, templates are updated by more recent examples in order to make the DTW-based recognizer adaptive.

6.7 Experiments

We evaluate realistic operational VUI procedures pertaining to a home automated setting in which speech-impaired users train the VUI. The explained procedures are compared in three experiments. In the first, several aspects are verified such as the use of a forgetting factor, the adjustment of GMM parameters by the transformation proposed in section 6.5.5 and the aid of different initialization procedures. In the second experiment, we compare the learning curve of incremental VUI learning against batch learning procedures, in addition to mixed procedures and DTW. In the third experiment, the adaptive capacity of these procedures is tested for sustained changes in user's voice.

6.7.1 Setup

Speech corpus

The DOMOTICA-3 database [3] contains Dutch, dysarthric speech commands that are typical to home automation. The dataset consists of recordings of speakers that also participated in the collection of the DOMOTICA-2 dataset used in earlier evaluations (see [25] and [12]). First, naturally evoked commands were collected from different users. Lists were composed of these commands. These lists were read repeatedly by multiple dysarthric speakers and led to the DOMOTICA-2 and DOMOTICA-3 dataset collection. The list number and some speaker characteristics such as gender, the total number of utterances ($\boldsymbol{N}$), the number of different commands (***commands***) and the intelligibility scores (***Intel. score***) [26] are listed in Table 6.2. The lists contained 27 commands, but some speakers received reduced lists of 10 commands. An intelligibility score above 85 is considered as normal whereas a score below 85 is considered as impaired. Intelligibility scores are missing for children with personal identification (Pid) 31 and 37 because the instrument in [26] is not designed for child voices. Dysarthria

was related to different pathologies such as spastic quadriparesis and multiple sclerosis.

list	*Pid*	*gen-der*	*N*	*com-mands*	*Intel. score*	*Pid*	*gen-der*	*N*	*com-mands*	*Intel. score*
1	43	♀	133	10	89.4	46	♀	97	10	74.9
4	32	♀	49	23	65.6	35	♀	282	27	72.3
5	48	♂	170	10	85.8	30	♂	222	27	69
6	17	♀	349	27	88.6	28	♀	212	27	73.1
8	31	♂	233	27	-	37	♂	171	10	-
2	34	♀	335	27	79.9	41	♀	144	27	66.7
1	29	♂	181	25	73.6					
3	33	♂	113	10	66.1					
9	44	♂	164	27	93.9					

Table 6.2: *Participants in* DOMOTICA-3

Evaluation procedure

The performance of the different procedures was evaluated on a test set that was set apart. It contained one randomly selected exemplar of each unique command. The remaining utterances served as training set. Ten folds were created and each fold presented sentences in a different permuted sequential order of the training utterances and a different test set. In order to evaluate incremental learning, training sets increased with epochs of 10 utterances ($O_i = 10$). Evaluation is based on recognition *F-scores* of semantic values in the test set.

Parameters and initialization

We used MFCC features and the spectral energy including the first and second derivative leading to $p = 42$ feature dimensions in total. Silence frames were removed by using a voice activity detection and mean and variance normalization was applied. $K = 50$ Gaussians was chosen which yielded the best performance for a vocabulary of $R = 29$ semantic entities in the experimental preparation phase.

We stacked four sets of HAC features with delays $\tau = 2, 5, 9$ and 20 resulting in 4×50^2 entries for each utterance-based acoustic representation. These delays have been used in other studies [12]. Each delay-dependent HAC set was treated as a separate multinomial distribution. The semantic multinomial stream was normalised and scaled to have the same L1-norm as the acoustic multinomial streams. Similarly, the semantic part of $\mathbf{W}$ had the same L1-norm as the acoustic part. The columns of $\mathbf{W}$ were normalised to 1.

In addition to the $R = 29$ columns in $\mathbf{W}$, a few extra $\mathbf{W}$-columns, $D = 5$, were added in order to model filler words. This proportion was constant for all

experiments. Each column in $\mathbf{H}$ was initialised as an uniform vector with the L1-norm equal to the L1-norm of the respective columns in the data matrix (see Eq. 6.35 to 6.37). The acoustic part of $\mathbf{W}$ was initialised with uniformly distributed random entries. The semantic part of $\mathbf{W}$ was initialised as follows,

$$\mathbf{W}_0 = \left[\frac{1}{2}\mathbf{I}^{(R\times R)} + \delta \qquad \frac{1}{2R}\mathbf{1}^{(R\times D)} + \mathbf{G}^{(R\times D)} \right]$$

with $\mathbf{I}$ the identity matrix and $\mathbf{1}$ a matrix of ones, both multiplied with $1/2$ in order to reserve 50% for the acoustics. δ is an arbitrary small constant larger than zero and $\mathbf{G}$ is a random matrix of appropriate size —dimension are specified in parentheses aside —drawn from the uniform distribution between 0 and 10^{-4}.

Hyperparameters $\xi_{ij}, \iota_{fj}, \lambda_k, \nu_k$ and α_k are set to $1, 5, 1, 43$ and 30000, respectively. Informative priors $\iota_{fj} = 5$ are chosen in order to avoid that columns of less frequent semantic entities are cultivated by more frequent ones after a few epochs, whereas the informative priors α_k are chosen to prevent that mixture proportions adapt to utterance-based statistics instead of data-based statistics. GMM parameters are initialised as follows: $\omega_k = 1/50, \mathbf{\Sigma}_k = \mathbf{I}$ and all μ_k are randomly selected points on the unit hypersphere surface or adopted from CGN clusters, depending on the initialization procedure at hand.

In the experiments, local dissimilarity is based on the cosine similarity between two feature vectors after mean and variance normalization [27]. If $\mathbf{x}_a$ and $\mathbf{x}_b$ are two mean and variance normalised vectors, then their local dissimilarity is defined as $d(\mathbf{x}_a, \mathbf{x}_b) = 1 - \frac{\mathbf{x}_a^T \mathbf{x}_b}{||\mathbf{x}_a|| ||\mathbf{x}_b||}$. In the DTW-based procedure, the last six spoken commands guided by the same unique semantic input were held as DTW templates. These templates were continuously updated with the new encountered examples. In decoding, the most similar template was choosen and the corresponding semantic vector was selected as prediction. This prediction is compared with the semantic annotation of the decoded utterance in the test set. This comparison allowed us to evaluate DTW on the same terms as all the other procedures. In a prior evaluation where we compared five against six retained example templates, we found small gains going from five to six. Therefore, we did not test more than six example templates per command.

6.7.2 Experiment 1

Setup

GMM adaptation induces changes in the first layer. Since these Gaussians are used as a codebook, these changes invalidate the acquired NMF representations

that are based on an old GMM. The proposed transformation in Eq. 6.49 reconstructs the NMF representations with respect to the developing Gaussians. VUI learning with and without the use of the transformation was compared for the full incremental procedure, that is "AI-GMM, AI-NMF". Additionally, incremental procedures with and without forgetting factor were evaluated. For this, a forgetting factor: $\eta, \gamma = 1$ and $\eta, \gamma = 0.95$ was chosen. A forgetting factor of 0.95 with epochs of 10 utterances corresponds with a forgetting factor of $0.995 \approx \sqrt[10]{0.95}$ for epochs containing a single utterance as depicted in Figure 6.1. The last variable of interest is the initialization of the Gaussian means: drawn randomly from the surface of a unit hypersphere, or initialised with the cluster means acquired by applying the k-means algorithm on 500,000 randomly selected frames from Corpus Gesproken Nederlands (CGN) [28]. This corpus contains Dutch spoken interviews and news broadcastings.

We evaluated the performance of these three variables with binary conditions in a fully crossed experiment and repeated each combination of these variables 10 times, using each time a different order of the utterances. The results were split into two groups of training sets: one group contained training sets of sizes smaller than 100 utterances which are listed in the middle column of Table 6.3, whereas the second group contained sets larger or equal to 100 utterances and listed in the third column of Table 6.3.

Results

In Table 6.3, the contrasts are listed for each group. Only the use of the transformation $\mathbf{T}$ seemed to yield a significant difference. The average gain was 3,3 % and 6,1% absolute improvement for the group of small training sets and the group of the large training sets, respectively. The performance drop by applying a forgetting factor was not significant and initialization with CGN yielded a non-significant improvement of 2.9% and 1.9% for each respective group of training sets. Based on these results, all incremental GMM procedures in the following experiments were fitted with CGN-based initialization and made use of the transformation expressed by Eq. 6.49. We used forgetting factors $\eta, \gamma = 0.95$ in the baselines of the following experiments.

6.7.3 Eperiment 2

Setup

The VUI learning curves of the procedures are evaluated. The learning curve provides a measure of the learning rate by which vocal expressions are acquired.

Learning examples average F-score	< 100 67%		≥ 100 85%	
contrast	**Δ (%)**	***stdev***	**Δ (%)**	***stdev***
$\eta, \gamma = 1$ - $\eta, \gamma = 0.95$	-0.4	0.7	-0.7	0.7
with **T**- without **T**	*9.4*	*4.1*	*9.5*	*3.1*
CGN_init - rand_init	2.3	2.1	0.9	2.3

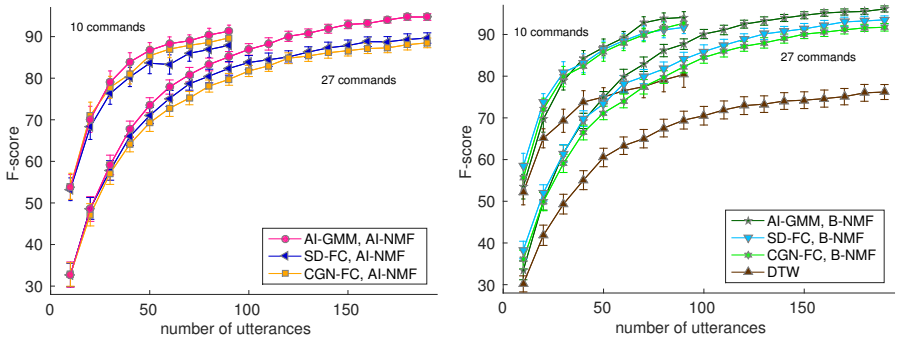
Table 6.3: *The average effect of the manipulations: without a forgetting factor against a forgetting factor, with the use of* **T** *against without* **T***, and initialization with CGN versus random.*

Additionally, in case of large training sets, the learning curve levels off and provides a measure of the asymptotic performance that can be reached.

Results

The learning curves of the memoryless procedures are depicted in Figure 6.2a, whereas the learning curves of the procedures requiring increasing data storage are depicted in Figure 6.2b. The x-axis represents the incrementally growing number of utterances in the training set. The longer curves include speakers with 27 different commands and more than 190 training utterances in total (see Table 6.2). These speakers have Pid $17, 28, 30, 31, 34$ and 35. The intelligibility scores range from 69 to 88.6 and was 76.6 on average. The shorter curves include speakers with Pid $33, 37, 43, 46$ and 48 who only spoke 10 different commands. The intelligibility scores in this group range from 66.1 to 89.4 and was 79.0 on average.

The graphs are especially useful to compare the different codebook procedures because the NMF layers are all the same in each separate panel. The best memoryless procedure is "AI-GMM, AI-NMF" displayed in Figure 6.2a with circle-shaped markers. For this procedure, the group with 10 different commands reached an F-score of 91.3% on average for training sets of 90 learning examples, whereas the other group reached an F-score of 94.7% on average for 190 learning examples. In Figure 6.2b, a similar pattern of results is displayed with respect to the procedures in the clustering layer. The "AI-GMM, B-NMF" procedure, marked with a five pointed star, reached the highest end scores with 94.1% and 96.1% for the short and longer curve, respectively. The short curves rise steeper than the longer ones possibly because of the more restricted vocabulary. Although the differences between the longer curves are clearly visible, a clear pattern of differences was not visible for the shorter ones. Nevertheless, for the longer curves it can be seen that incrementally learned codebooks outperform



a) Data memoryless procedures b) DTW and procedures storing data

Figure 6.2: *The VUI learning curves for the first 190 utterances, averaged over speakers. The errorbars are the average standard errors of the speakers. Individual end scores are presented in Table 6.4*

codebooks trained on pre-recorded user data or CGN. The differentiation of these curves starts at about 50 training examples and becomes significant at about 80 to 90 training examples for the longer ones.

In Table 6.4, the final F-scores for each individual is listed for "AI-GMM, AI-NMF", "AI-GMM, B-NMF" and "SD-FC, B-NMF". When comparing F-scores of the two procedures building further on incremental GMM "AI-GMM", i.e. columns six and seven in Table 6.4, it can be seen that batch NMF was performing better than incremental NMF with an average difference of 1.9%. Batch NMF learning together with speaker-dependent codebooks "SD-FC, B-NMF" as used in [12] and perfoms at the same level as the incremental procedure "AI-GMM, AI-NMF".

All proposed VUI procedures outperformed DTW. An important observation in Table 6.4 is the influence of the vocabulary size: although learning curves for small vocabulary had a steeper rise, this rise would correspond closely with the rise of the longer curves if learning was evaluated with respect to the number of learning examples per command listed in column five of Table 6.4.

Pid	*Intel. score*	*Training set size*	*Com-mands*	*Command examples*	*F-score (%),* $\eta = \gamma = 0.95$		
					AI-GMM AI-NMF	AI-GMM B-NMF	FC-SD B-NMF
17	88.6	322	27	11.9	99.6	**100**	99.4
28	73.1	185	27	6.9	95.4	**96.9**	94.5
29	73.6	154	25	6.2	96.7	**97.5**	92.0
30	69.0	195	27	7.2	94.8	**96.2**	92.9
31	-	206	27	7.6	91.5	**92.1**	86.2
32	65.6	26	23	1.1	**65.7**	65.1	64.4
33	66.1	103	10	10.3	68.5	**85.5**	79.0
34	79.9	335	27	12.4	**98.3**	**98.3**	97.9
35	72.3	265	27	9.8	96.2	**97.2**	95.3
37	-	161	10	16.1	91.5	**94.0**	93.0
41	66.7	117	27	4.3	96.0	**97.6**	96.2
43	89.4	123	10	12.3	**100**	**100**	99.5
44	93.9	137	27	5.1	99.4	**100**	99.4
46	74.9	87	10	8.7	98	**99.5**	99.0
48	85.8	160	10	16.0	**100**	**100**	98.0

Table 6.4: *Individual F-scores for different procedures using all available data.*

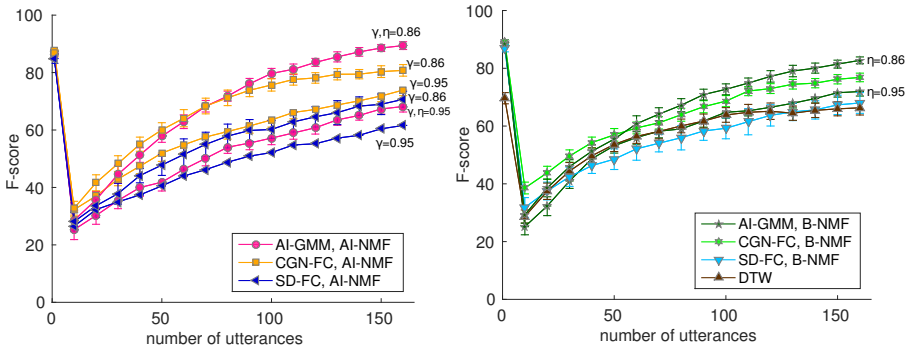
6.7.4 Experiment 3

Setup

The adaptive capacity of the procedures was evaluated for changes in user's vocal characteristics. Such changes emerge in users with a progressive disease during their life span. Since the voice recordings are snapshots of two consequtive moments over a time span of one half year resulting in the DOMOTICA-2 and DOMOTICA-3 data sets, we were not able to track this kind of regression in the speaker's voice. Therefore, the utterances of one user were appended to the utterances of another one with the same gender and command list number. The pairs of participants are listed in the first six rows of Table 6.2. All utterances in the appended lists were administered to the learning algorithms as if the utterances were produced by the same user. We investigated which learning procedure was able to adapt to the new vocal characteristics by evaluating the recovery from the user change. For this, we compared adaptive incremental procedures with forgetting factors equal to 0.86 and 0.95. Considering epochs of one utterance, a forgetting factor of 0.985 as depicted in Figure 6.1, corresponds with a factor of 0.86 using epochs of 10 utterances.

Results

In Figure 6.3, the average F-scores for the end speakers with Pid $28, 30, 35$ and 37 of the user pairs are plotted against the first 160 utterances following



a) Data memoryless procedures b) DTW and procedures storing data

Figure 6.3: *Adaptation demonstrated by the different VUI learning curves averaged over speakers for the first 160 utterances following the user change. The errorbars are the standard errors. Individual end scores are presented in Table 6.5*

the user change. The two excluded end users in the graph had less than 160 utterances, nevertheless, their end scores are listed in Table 6.5. The NMF incremental learning procedures are depicted in the left panel whereas the NMF batch learning procedures are depicted in the right panel. The F-scores at the origin correspond with the average ending scores of the users preceding the user change. The drop in performance between 0 and 10 utterances results from the user change. From there, the performance recovers at different rates for different procedures. For all procedures involving incremental learning, two curves with the same markers and colors are depicted with their forgetting factors 0.86 or 0.95 displayed aside. Contrary to the fast learning experiments, the incremental procedures performed better than the batch learning procedures. The full incremental procedure "AI-GMM, AI-NMF", depicted in the left panel by circle-shaped markers, reached the higest score of 89.4% at 160 utterances. The second best procedure was the NMF batch learning procedure backed up with an incremental GMM procedure "AI-GMM, B-NMF", reaching a score of 83.2% at 160 utterances. This curve is depicted in the right panel with five-pointed star markers. Clearly, this procedure had a considerable drop compared with the full incremental procedure. However, some adaptation was achieved through the incremental GMM training procedure as can be seen by the different performances for different forgetting factors η in the clustering layer. Overall, when considering the curve pairs of the incremental procedures, the steepest rise is obtained for the curve guided by the strongest forgetting factor. For instance, "CGN-FC, AI-NMF" reached a score of 76% at 100 utterances

by using a forgetting factor of 0.86; this score was 12.4% higher than the same procedure using a forgetting factor of 0.95. This relative performance gap was the largest for the "AI-GMM, AI-NMF" procedure with incremental learning at both layers. Note also that procedures using speaker-dependent clusters "SD-FC", performed worse than procedures using CGN-based clusters "CGN-FC". The speaker-dependent training material involved only the preceding speaker.

More detail is presented in Table 6.5. In this table, the end scores of incremental procedures using the stronger forgetting factor are presented together with batch procedures. End scores comparable with the ones in Experiment 2 are only achieved for the fully adaptive procedure: the 'AI-GMM, AI-NMF" with γ and η equal to 0.86. The end scores of speaker $46, 35, 30, 28$ and 37 in Table 6.5 are approaching the respective end scores in Table 6.4. Another interesting observation is the overall good performance for all procedures of speaker pairs $43 \rightarrow 46$ and $32 \rightarrow 35$. The training set size of the first speakers counted 123 and 26 utterances, respectively, strongly contrasting the 335 and 322 utterances of the first speakers 34 and 17, respectively. The more utterances prior to the user change, the stronger the priors and the more new utterances needed to unlearn the old models.

list	***Pid's***	***F-score (%)***, $\eta = \gamma = 0.86$				
		AI-GMM AI-NMF	AI-GMM B-NMF	FC-CGN AI-NMF	FC-CGN B-NMF	DTW
1	43→46	**96**	93	92.5	95.5	94.0
2	34→41	**84.9**	60.1	83.6	76.7	75.2
4	32→35	95.1	**96.5**	91.1	94.1	83.8
5	48→30	**93.9**	80.2	72.8	65.6	60.4
6	17→28	**92.3**	76.3	80.7	68.9	78.1
8	31→37	88	**91**	86	87.5	53.5

Table 6.5: *Individual F-scores for different procedures using all available data.*

6.8 Discussion

It is shown that incremental learning procedures based on MAP estimation require slightly more training data to achieve the same accuracy than their batch learning variants. MAP estimation at the clustering layer leads to better codebooks than fixed codebooks based on CGN or based on speaker-dependent prior recorded data. It is thus a considerable advantage to use the most recent data for model estimation. A tentative explanation for faster batch learning is that the more data provided as a whole, the more irrelevant features are factored out leading to sparser representations. Whereas batch learning leads to sparser representations, incremental MAP updates keep track of sufficient statistics

which are a accumulation of all features: relevant and irrelevant acoustic features that co-occurred with semantic entries that were rather presented in isolation. If this assumption is true, then sparsity inducing priors might improve NMF MAP estimation. This assumption is subject to future research. From the perspective of the targeted application, the small drop in performance should be balanced against memory requirements.

The implementation of incremental MAP estimation on both layers is challenging because changes in the Gaussians require adjustments in the NMF representations in order to achieve proper decoding. It is shown in Section 6.7.3 that the introduced transformation is useful to achieve this goal. If the data is stored or if fixed codebooks are used, the transformation is not required. Only the full incremental procedure operates with this transformation between successive epochs.

The incremental procedures demonstrated better adaptation performance than our DTW implementation updating its reference templates online. Exhaustive Bayesian frameworks exists from which a straightforward MAP adaptation procedure could be applied to our GMM and NMF model. Conversely, adaptation in a template based vocal interface is not a straightforward procedure. One of the main advantages of the statistical NMF-based approach is that it parses utterance automatically based on statistical recurrency of the data. The parsing corresponds with the semantic content as it is regularised by the semantic supervision included in the utterance-based input vectors. Utterances are learned as a whole in the DTW procedure. A DTW procedure that learns keywords by segmented input vectors might demonstrate better performances. However, this would require an enrollment phase in which the user provides spoken keyword learning examples to the VUI.

Batch learning procedures learn slightly faster, but the use if incremental procedures is most advantageous if adaptation is required to changes in speech characteristics. This will probably prove to be more robust as well since the acoustic features are learned in the environment of the end user. If forgetting factors are chosen correctly, strong recovery is obtained. The performance levels after recovery in the third experiment approach the performance levels in the second experiment. These procedures outperform the batch learning procedures in a rather compelling way. However, if forgetting factors are improperly chosen, adaptation is suboptimal for small training sets. This finding raises new issues such as the selection of an appropriate forgetting factor. A dynamic forgetting value that weakens priors to an appropriate extend with regard to changes in user's behaviour is a promising direction of future research. More research is also required to find a good detection of possible acoustic changes opposing those such as non-persistent changes caused by a cold to name one example.

6.9 Conclusion

Overall, the performance of the incremental procedures are acceptable and feasible for VUI applications dealing with small vocabularies. They outperformed a DTW procedure using six templates per command. Similar to the DTW approach that builds or selects templates from scratch, the full incremental VUI learning approach is, to the best of our knowledge, the first statistical model-based approach that builds its ASR models from MFCC features and semantic content. Although NMF batch learning provides slightly faster learning, the rate of adaptation is considerable faster for incremental learning given a proper forgetting factor. Thus if adaptivity is required, if memory is restricted or memory control is needed, then the full incremental procedure is a viable and feasable solution. All its practical advantages make it suitable for many hardware platforms.

6.10 References

[1] J. Driesen and H. Van hamme, "Modelling vocabulary acquisition, adaptation and generalization in infants using adaptive Bayesian PLSA," *Neurocomput.*, vol. 74, pp. 1874–1882, May 2011. pages 150, 155, 160, 161

[2] G. Hinton, L. Deng, D. Yu, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. S. G. Dahl, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, pp. 82–97, November 2012. pages 151

[3] J. F. Gemmeke, B. Ons, M. Tessema, J. Van de Loo, G. De Pauw, W. Daelemans, J. Huyghe, J. Derboven, L. Vuegen, B. Van Den Broeck, and H. Van hamme, "Self-taught assistive vocal interfaces: An overview of the aladin project," in *Proceedings of Interspeech*, 2013. pages 151, 166

[4] J. Driesen, *Discovering words in speech using matrix factorization.* PhD thesis, K.U.Leuven, ESAT, July 2012. pages 151

[5] B. Ons, J. F. Gemmeke, and H. Van hamme, "Fast vocabulary acquisition in an NMF-based self-learning vocal user interface," *Computer Speech & Language*, vol. 28, no. 4, pp. 997–1017, 2014. pages 151, 152, 153, 154, 155, 165

[6] J. F. Gemmeke, S. Sehgal, S. Cunningham, and H. Van hamme, "dysarthric vocal interfaces with minimal training data," pages 151, 152

[7] M. Hawley, P. Enderby, P. Green, S. Cunningham, S. Brownsell, J. Carmichael, M. Parker, A. Hatzis, P. O. Neill, and R. Palmer, "A speech-controlled environmental control system for people with severe dysarthria," *Medical Engineering & Physics*, vol. 29, no. 5, pp. 586–593, 2007. pages 151

[8] M. S. Hawley, S. P. Cunningham, P. D. Green, P. Enderby, R. Palmer, S. Sehgal, and P. O'Neill, "A voice-input voice-output communication aid for people with severe speech impairment," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 21, no. 1, pp. 23–31, 2013. pages 151

[9] Z. Xianglilan, S. Jiping, and L. Zhigang, "One-against-all weighted dynamic time warping for language-independent and speaker-dependent speech recognition in adverse conditions," *PLoS ONE*, vol. 9, p. e85458, 02 2014. pages 152

[10] W. H. Abdulla, D. Chow, and G. Sin, "Cross-words reference template for dtw-based speech recognition systems," in *TENCON 2003. Conference on Convergent Technologies for the Asia-Pacific Region*, vol. 4, pp. 1576–1579, IEEE, 2003. pages 152

[11] L. Broekx, K. Dreesen, J. F. Gemmeke, and H. Van hamme, "Comparing and combining classifiers for self-taught vocal interfaces," in *Proc SLPAT*, (Grenoble, France), pp. 21–28, 2013. pages 152

[12] B. Ons, J. F. Gemmeke, and H. Van hamme, "The self-taught vocal interface," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2014, no. 1, p. 43, 2014. pages 152, 153, 154, 155, 165, 166, 167, 171

[13] V. Roy, S. Madhvanath, S. Anand, and R. Sharma, "A framework for adaptation of the active-dtw classifier for online handwritten character recognition," in *10th International Conference on Document Analysis and Recognition, 2009. ICDAR '09.*, pp. 401–405, July 2009. pages 152

[14] M. Matassoni, R. Astudillo, A. Natsamanis, and M. Ravanelli, "The dirha-grid corpus: baseline and tools for multi-room distant speech recognition using distributed microphones," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014. pages 152

[15] B. Lecouteux, M. Vacher, and F. Portet, "Distant speech recognition in a smart home: Comparison of several multisource asrs in realistic conditions," *Proc Interspeech*, pp. 2273–2276, 2011. pages 152

[16] H. Christensen, I. Casanuevo, S. Cunningham, P. Green, and T. Hain, "Homeservice: Voice-enabled assistive technology in the home using cloud-based automatic speech recognition," in *Proc SLPAT*, (Grenoble, France), pp. 29–34, 2013. pages 152

[17] H. Van hamme, "HAC-models: a novel approach to continuous speech recognition," in *Proc. Interspeech*, (Brisbane, Australia), pp. 255–258, 2008. pages 153

[18] J. Driesen, L. ten Bosch, and H. Van hamme, "Adaptive non-negative matrix factorization in a computational model of language acquisition," in *Proc. INTERSPEECH*, pp. 1731–1734, 2009. pages 155

[19] M. Shashanka, B. Raj, and P. Smaragdis, "Probabilistic latent variable models as nonnegative factorizations," *Computational intelligence and neuroscience*, vol. 2008, 2008. pages 155

[20] E. Gaussier, C. Goutte, K. Popat, and F. Chen, "A hierarchical model for clustering and categorising documents," in *Advances in Information Retrieval*, pp. 229–247, Springer, 2002. pages 155

[21] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *Speech and audio processing, ieee transactions on*, vol. 2, no. 2, pp. 291–298, 1994. pages 155, 157, 158, 159, 160

[22] J.-T. Chien and M.-S. Wu, "Adaptive bayesian latent semantic analysis," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, pp. 198–207, Jan 2008. pages 155, 156

[23] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–38, 1977. pages 157, 158

[24] T. Hofmann, "Probabilistic latent semantic analysis," in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pp. 289–296, Morgan Kaufmann Publishers Inc., 1999. pages 159

[25] B. Ons, N. Tessema, J. van de Loo, J. F. Gemmeke, G. De Pauw, W. Daelemans, and H. Van hamme, "A self learning vocal interface for speech-impaired users," *Proc SLPAT 2013*, pp. 1–9, 2013. pages 166

[26] C. Middag, *Automatic Analysis of Pathological Speech*. PhD thesis, Ghent University, Belgium, 2012. pages 166

[27] M. Ferrarons, X. Anguera, and J. Luque, "Flexible stand-alone keyword recognition application using dynamic time warping," in *Advances in Speech and Language Technologies for Iberian Languages*, pp. 158–167, Springer, 2014. pages 168

[28] N. Oostdijk, "The spoken dutch corpus. overview and first evaluation.," in *Proc. LREC*, (Genoa, Italy), 2000. pages 169

Chapter 7

Conclusion

7.1 Summary

In this dissertation, we aimed at a fully adapted VUI model that learns the vocalizations and the words from the user. To this end, we introduced and adapted machine learning techniques, which are able to learn from demonstrations provided by the user in a realistic user environment with incrementally available speech data. Important performance measures were learning rate and the accuracy or the F1-score at which performance peaks and levels off. The success of the VUI results were evaluated against the following research goals: the ability to learn from the user, to learn from a few learning examples and to learn from non-standard speech such as dysathric speech. Other research goals are to learn incrementally and to enable adaptation by which users can choose and change their words or pronunciations.

The research developed from the NMF word learning model in [1] towards a personalised model that learns incrementally from a few learning examples and that adapts to changes in the practices of the user through MAP procedures and forgetting factors. We give a concise overview of the contributions in each chapter and give our focus on future research.

7.2 Contributions and possible directions for future work

The main contributions to the computational model are the introduction of the MAP procedures in **chapter 6**. The MAP approach of the GMM was based on the work of [2]. We adapted the approach to learn incrementally and we derived new equations by adding a forgetting factor to the GMM auxiliary function. The MAP approach of incremental NMF was based on [3] and [4]. We derived new equations for stream combination and added stream weights to incremental NMF learning. The main contribution is the coupling between incrementally modified GMM and the adaptation of old NMF representations to new updated ones. This update is done by a transformation in the space of the NMF hyperparameters. Without this coupling, incremental learning starting from scratch would simply not work. This is demonstrated in the first experiment of the respective chapter. Besides this main contribution, we make contributions in **chapter 5** by introducing a decision process and in **chapter 4** by a practical definition of smoothing.

The main contribution to research methodology in the area is the conduction of well-designed experiments on small data sets. This is a non-trivial problem. The scarce data in for example **chapter 5**, hampers a reliable measure of performance for small training sets and held-out test sets. To this end, we used procedures such as cross-validation and we introduced a new procedure to partition the data in matched blocks based on minimizing the Jensen-Shannon divergence of the slot value distributions across these blocks. In **chapter 5**, we also adapted non-parametric bootstrap procedures to estimate confidence intervals for the learning curve of the average dysarthric speaker obtained by non-parametric regression methods ([5, 6]). These statistical methods allow to make sense of small differences between different experimental conditions. Well-designed experiments are important as they allow us to make conclusions and closures on particular research issues.

In each chapter, we made a few contributions and left a few relevant questions unexplored. In **chapter 1**, we discussed the importance of our aims in the light of social, economical, historical and philosophical aspects. We explained the followed approach and the motivation to follow this approach in order to fulfil the research goals. We gave an overview of NMF uses and the feature extraction techniques for NMF. A few of these techniques were introduced in chapter 3.

In **chapter 2**, we tested four realistic scenarios in which inconsistent semantic labels emerge during demonstrations. The label noise was randomly added and uncorrelated. Uncorrelated noise means that the probability of a confusion is the same for every pair of labels. In the real-world learning environment, label

noise might be correlated. For example, if multiple light switches are embedded in the same switching unit on the wall, then more confusion can be expected between those switches that are co-located. Occasionally, pairwise confusion is not symmetric and one label might be more dominant than the other. The investigations of these realistic scenarios together with the incorporation of context information are potential research extension to this chapter. We have shown for the first time that the approach is inherently very robust against randomly added label noise and consequently there should be no priority on making it more robust.

In **chapter 3**, we conducted an exhaustive examination of different processing flows that let us make stepwise improvements from speaker-independent models to speaker-dependent models. One of the contributions that needs further investigation is fast learning by using HAC features based on phone posteriorgrams. The advantage of the phone-based HMM is that it comprises a lot of information on human speech and allows for a concise decoding format. However, training a speaker-dependent HMM requires supervision and this is hard to obtain in the VUI-user training context. Moreover, whether HMMs are useful in the case of dysarthric speech is still an open issue. An alternative direction of further research is to adopt the procedure in [7] or the HMM MAP approach in [2] to learn phone-like HMM-based subword units from the speech of the user. The development of speaker-dependent HMM-based subword units might require more learning examples, but the scalability of the model would improve allowing for command-and-control applications with a lexicon of a larger scale. One of the reasons that this research path was not pursued was the size of the Domotica corpora starting from a few utterances for each speaker during the first years of the project, up to one or two hundred utterances at the end of the project. Unsupervised learning of HMM acoustic subword units on small datasets is probably not very succesful and therefore left for further generations of this VUI technology. The most important contribution is that we gain confidence that models that learn and build their representations from the user are feasible and demonstrate better results in the long run than of-the-shelve speaker-independent ASR components.

In **chapter 4**, we introduced two methods to improve learning from scarce data: smoothing of posteriorgrams and constraining the number of free parameters. The smoothing of the data was a promising technique: performance improved for scarce training data and did not decrease for large data sets. In the preceding chapter, it was shown that multiple input streams are easily combined in NMF. An alternative procedure that could have led to a similar effect as smoothing is a parallel presentation of multiple streams that encode the same data. For example, the use of multiple different codebooks counting 100 clusters each on which multiple data streams are based and presented could reduce overfitting.

Such a redundant, but also richer presentation might even improve performance for large data sets and gather more information for data sets of a few utterances. This will affect the learning speed. The effect on accuracy of multiple parallel code books in NMF was investigated in [8] and promising results were obtained. So, it would have been instructive to have it implemented in this chapter. An additional contribution of this chapter is that we demonstrated that the influence of the semantic vectors on the factorization of the acoustic vectors should be weak (as opposite to restricted). This weak influence facilitates the self-organizing ability of NMF to decompose the data in patterns that provide better accuracy for larger data sets.

In **chapter 5**, we evaluate learning in the vocal user interface using two corpora, one containing recordings of dysarthric spoken commands related to home automation, and one with recordings of normal speech with speakers playing the card game patience. For dysarthric speech, we demonstrated adequate learning of commands from a few learning examples. We demonstrated better performance for semantic frame structures with hierarchical and compositional layers. We described the decision process in which activation values are compared against each other and propagated to each level of the hierarchical semantic frame structure. However, there was no experimental validation of the decision process since the focus of the article was on the evaluation of the ALADIN system on dysarthric speech. Moreover, we were standing on the threshold of launching a complete new approach, involving group sparsity [9]. In group sparsity, the activation of one group competes with the activations of other groups. A semantic value would become part of a group containing all semantic values of that frame, and group sparsity would lead to the predominance of one group (frame) above the others. The development of group sparsity is a new direction of research that could replace the decision stage of the current model. The hierarchical decision process could be evaluated against group sparsity on learning semantic frames in further research.

In **chapter 6**, we introduced a VUI model with incremental and adaptive learning algorithms that develop its Gaussian components and its lexicon during usage without prior knowledge. The development of the Gaussian components on the up-to-date speech commands of the user yielded significantly better performance than Gaussians trained on speaker-dependent prior recordings of the user. This is one of those findings that justifies our attention on learning from the operational context. In the experiments, we compared the new VUI model with other incremental learning procedures and their complementary batch learning variants. Learning is a dynamic process in the sense that representations should continuously adapt. Our new VUI model demonstrated fast adaptation. A valuable extension of research is to add the ability to grow by adding new Gaussian components or splitting existing ones, or by increasing

the model order of the NMF learning model as more semantic frames emerge or new synonyms or concepts are used. The automatic regulation of the model order selection (the number of latent components) and the search for robust measures that signal the use of new words or new commands is a challenge.

The introduced and adapted techniques in these chapters are valuable improvements to the VUI model. They make the VUI more pleasant to use by the personalised acoustic decoding and personalised lexicon. In this dissertation, we mainly focused on the development of the acoustic model and the vocabulary in the learning context. We gave some thoughts on further research such as the use of group sparsity, the scaling towards larger vocabulary applications and enabling models to grow. These new research directions could lead to a speech interface that could be hooked up to a wide range of applications and that caters for users with non-standard speech or for normal speaking users who like to use their own words.

In this research, we learned that speech recognition building up from no or limited prior knowledge, thus developing models by listening to and learning from the user, is a feasible approach. In an application with a limited lexicon of no more than a hundred words, this approach is able to reach a ceiling accuracy close to 100% in a reasonable time requiring a limited user effort that involves one or a few learning examples per concept. These accuracies are obtained on condition that the system learns from user- and usage-dependent data, and this on different levels such as the acoustic units and the wordlike NMF patterns that refer to the application-dependent concepts. Moreover, this approach is state-of-the-art for command-and-control applications involving severe dysarthric speech. It widens accessibility for those people for which common ASR is not a viable solution.

7.3 References

[1] J. Driesen, J. F. Gemmeke, and H. Van hamme, "Weakly supervised keyword learning using sparse representations of speech," in *Proc. ICASSP*, (Kyoto, Japan), pp. 5145–5148, 2012. pages 181

[2] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *Speech and audio processing, ieee transactions on*, vol. 2, no. 2, pp. 291–298, 1994. pages 182, 183

[3] J.-T. Chien and M.-S. Wu, "Adaptive bayesian latent semantic analysis," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, pp. 198–207, Jan 2008. pages 182

[4] J. Driesen, L. ten Bosch, and H. Van hamme, "Adaptive non-negative matrix factorization in a computational model of language acquisition," in *Proc. INTERSPEECH*, pp. 1731–1734, 2009. pages 182

[5] W. S. Cleveland, "Robust locally weighted regression and smoothing scatterplots," *Journal of the American statistical association*, vol. 74, no. 368, pp. 829–836, 1979. pages 182

[6] W. S. Cleveland and S. J. Devlin, "Locally weighted regression: an approach to regression analysis by local fitting," *Journal of the American Statistical Association*, vol. 83, no. 403, pp. 596–610, 1988. pages 182

[7] S. Meng and H. Van hamme, "Joint training of non-negative tucker decomposition and discrete density hidden markov models," *Computer Speech & Language*, vol. 27, no. 4, pp. 969 – 988, 2013. pages 183

[8] S. Meng and H. Van hamme, "Coding methods for the NMF approach to speech recognition and vocabulary acquisition," in *In Proc. International Multi-Conference on Complexity, Informatics and Cybernetics: IMCIC*, (Florida, USA), 2011. pages 184

[9] R. Jaiswal, D. Fitzgerald, E. Coyle, and S. Rickard, "Shifted NMF with group sparsity for clustering NMF basis functions," in *Proc at 15th International Conference on Digital Audio Effects DAFx-12*, (York, UK), pp. 17–21, 2012. pages 184

List of Publications

Articles in International Journals

[1] Ons B., Gemmeke J.F., Van hamme H.,"Incremental adaptive learning in the self-taught vocal interface", IEEE Transactions on Audio Speech and Language Processing (Submitted).

[2] Ons B., Gemmeke J.F., Van hamme H., "The self-taught vocal interface", EURASIP journal on Audio, Speech and Music Processing, 2014:1:43, doi:10.1186/s13636-014-0043-4.

[3] Ons B., Gemmeke J.F., Van hamme H., "Fast vocabulary acquisition in an NMF-based self-learning vocaluser interface", Computer speech and language, vol. 28, no. 4, pp. 997-1017, July 2014.

[4] van de Loo J., Gemmeke J.F., De Pauw G., Ons B., Daelemans W., Van hamme H., "Effective weakly supervised semantic frame induction using expression sharing in hierarchical hidden Markov models", Computer speech and language (submitted).

Articles in International Conferences

[1] Gemmeke J.F., Ons B., Tessema N., Van hamme H., van de Loo J., De Pauw G., Daelemans W., Huyghe J., Derboven J., Vuegen L., Van Den Broeck B., Karsmakers P., Vanrumste B., "Self-taught assistive vocal interfaces: An overview of the ALADIN project", Proceedings 14th annual conference of the International Speech Communication Association (ISCA) - Interspeech 2013, pp. 2038-2043, August 25-29, 2013, Lyon, France.

[2] Ons B., Gemmeke J.F., Van hamme H., "NMF-based keyword learning from scarce data", Automatic speech recognition and understanding workshop - ASRU 2013, pp. 392-397, December 8-12, 2013, Olomouc, Czech Republic.

[3] Ons B., Tessema N., van de Loo J., Gemmeke J.F., De Pauw G., Daelemans W., Van hamme H., "A self learning vocal interface for speech-impaired

users", 4th workshop on speech and language processing for assistive technologies - SLPAT-2013, pp. 73-81, August 21-22, 2013, Grenoble, France.

[4] Ons B., Gemmeke J.F., Van hamme H., "Label Noise Robustness and Learning Speed in a Self-Learning Vocal User Interface." In Natural Interaction with Robots, Knowbots and Smartphones. Springer New York, pp. 249-259, 2014.

[5] Renkens V., Janssens S., Ons B., Gemmeke J.F., Van hamme H., "Acquisition of ordinal words using weakly supervised NMF." In Proc. IEEE Spoken Language Technology Workshop (SLT), pages 1-6, 2014.

[6] Walter O., Despotovic V., Haeb-Umbach R., Gemmeke J.F., Ons B., Van hamme H., "An evaluation of unsupervised acoustic model training for a dysarthric speech interface", Proceedings 15th annual conference of the International Speech Communication Association (ISCA) - Interspeech 2014, pp. 1013-1017, September 14-18, 2014, Singapore.

Technical Reports

[1] Tessema N., Ons B., van de Loo J., Gemmeke J.F., De Pauw G., Daelemans W., Van hamme H., "Metadata for Corpora PATCOR and Domotica-2", Technical report KUL/ESAT/PSI/1303, KU Leuven, ESAT, July 2013, Leuven, Belgium.

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF ELECTRICAL ENGINEERING (ESAT)
CENTER FOR PROCESSING SPEECH AND IMAGES (PSI)
Kasteelpark Arenberg 10 - box 2441
B-3001 Heverlee
bart.ons@esat.kuleuven.be